
La revue des
SHARPENTIERS
PC 7000
le compatible portable
N°15 - BIMESTRIEL - NOVEMBRE 85 - 30 F.
LA REVUE DES UTILISATEURS DE MICRO-ORDINATEURS ET POCKET-COMPUTERS SHARP





INFORMATIQUE

CONSEIL-SERVICE Location-Leasing

78, Bd Montebello, 59000 LILLE - Tél. 20 93 66 66
ouvert du lundi au vendredi de 9 à 12 h et de 14 à 18 h



MUSCLEZ VOS MZ...

Individuels 8 Bits

0. *MICRO ORDINATEURS*
Toute la gamme des MZ.

1. *MONITEURS*
Vert, Ambre, Couleur, H.R.

2. *IMPRIMANTES*
7, 9, 18 aiguilles, graphique, courrier, couleur, laser...
Rapide et lisible

3. *TABLE TRACANTE*
Couleur, multiplume...
Un croquis vaut mieux...

4. *DISQUES*
Interface disquette, lecteurs et disquettes.
le disque, c'est tout et tout de suite!

5. *DIVERS*
Carte RAM, Graphique 80B - Carte 80 colonnes MZ 700 - Interfaces CENTRONICS - Réseau de micros scolaires - Maintenance...

6. *FOURNITURES*
Rubans, papiers, disquettes...
Manuels BASIC, CPM, DOS, DBASE...



VOTRE MZ S'ENNUIE?

Monoposte 8 Bits	**Multiposte 16 Bits**
MZ 700, MZ 800 MZ 80B, MZ 3541	MZ 5600

7. *SYSTEMES D'EXPLOITATION*

MERCURE CPM EOS	MOS Mercure DOS MSDOS PROLOGUE

8. *LOGICIELS*
LOTUS, MULTILOG...
le calcul cool... MULTIPLAN!

9. *PROGICIELS*
Comptabilité, paie.
Stock-facturation.
Gestion de production.
Plus pénible que BASIC, tu meurs. D BASE ça existe...

OFFRE DE LANCEMENT EXPER 700
logiciel d'intelligence artifielle

Les solutions existent chez BECY! Consultez-nous...

Un CHOIX de PRIX, c'est BECY!

COUPON REPONSE à retourner à BECY avec timbre à 2,20 F

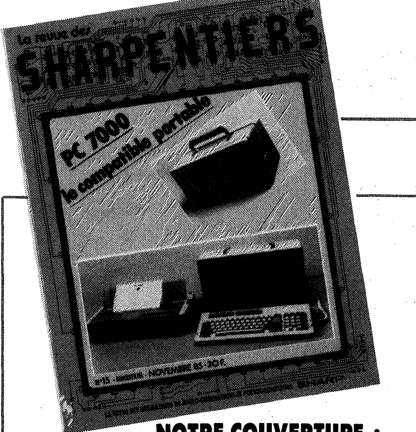
Je désire recevoir la documentation sur [0] [1] [2] [3] [4] [5] [6] [7] [8] [9]

Je commande le logiciel EXPER 700 ci-joint un chèque de 30 Francs.

Nom .... Adresse .... Signature :

....

Code Postal |_|_|_|_|_| ville ....



NOTRE COUVERTURE :
Léger, puissant, évolutif tel est le dernier né de la gamme des PC-SHARP. A ces atouts s'ajoute une totale compatibilité avec le système d'exploitation le plus universel du marché (voir nouveautés page 2).

CLUB DES SHARPENTIERS
151/153, avenue Jean-Jaurès
93307 AUBERVILLIERS CEDEX
Tél. : 48.34.93.44.

REDACTEUR EN CHEF :
Sylvain BIZOIRRE

REDACTEURS :
Luc BURELLER
Marc GIRONDOT
Jean-François VIGNAUD

SECRETAIRE DE REDACTION :
Graziella HAYET

Ont collaboré à ce numéro :
E. BERNARD, A. MORETTE, H. PANETTO, X. PIERSON, D. MAGNIN, B. KOKANOSKI, G. VINCENT, D. BEURRIER

PUBLICITE :
S. BIZOIRRE
Tél. : 48.34.93.44.

REALISATION :
IN QUARTO
19, rue Frédéric Lemaître
75020 PARIS

CRITERIUM SHARPENTIERS

Les règles en sont très simples mais peuvent rapporter gros aux SHARPENTIERS courageux :

1. Tout Sharpentier dont l'article, le programme ou l'astuce aura été publié dans le bulletin SHARPENTIER sera d'office inscrit au club, gratuitement, pour une nouvelle année.

2. A la parution de chaque bulletin, l'un de ces auteurs se verra offrir l'un des plus récents produits SHARP.

Vainqueur du numéro 14 : Mr. Gilbert VINCENT, auteur de trois superbes articles sur PC 1 211. Il gagne ainsi un PC 1 450.

Pour le numéro 16, il y a un PC 1 100 à gagner.

A vos claviers...

LE SHARPENTIER — SOMMAIRE N° 15

PC 7000
LE COMPATIBLE PORTABLE

Plutôt que de nouveauté, nous pourrions parler de révolution avec le PC 7000 car celui-ci introduit un nouveau concept de l'ordinateur personnel : le portable à écran vertical.

Le dernier né de la gamme « système » de SHARP se présente sous la forme d'une malette intégrant l'unité centrale et les 4 périphériques de base : Clavier, écran, disquettes et imprimante. Entièrement compatible IBM PC et XT (*), il fonctionne sous le système d'exploitations MS.DOS 2.11 (*).

En version de base, la capacité de sa mémoire vive : 384 Ko. suffit amplement pour faire tourner les plus volumineux logiciels existants. La particularité de son écran à cristaux liquides est d'être éclairé, de l'arrière, par un panneau électroluminescent. Cette technique élimine définitivement la fatigue visuelle engendrée par les classiques moniteurs vidéo. Affichant 25 lignes de 80 caractères, il autorise une résolution graphique de 640 x 200 points.

Les 2 unités de disquettes 5 pouces 1/4 sont montées sur « silent block » ; elles sont donc insensibles aux chocs et pratiquement inaudibles en fonctionnement. Compatibles aux formats IBM PC / XT (*), elles peuvent conserver jusqu'à 360 Ko. formattés.
L'imprimante qualité courrier fait partie intégrante de la malette ; elle possède 3 formats d'impression et 4 différentes polices de caractères.

Le clavier confirme la comptabilité IBM puisqu'il possède une disposition des touches identique à celle d'un IBM PC / AT (*).
Comptabilité encore puisque l'adaptateur pour écran couleurs au format IBM est intégrable dans l'unité centrale du PC 7000.

Un boitier d'extensions optionnel prend place sous l'unité centrale. Il intègre 3 connecteurs au format des cartes IBM / PC (*) et un disque dur d'une capacité de 10 millions de caractères.

La portabilité, la fiabilité unanimement reconnue des produits SHARP et la totale comptabilité avec la plus grande banque de logiciels 16 bits : 3 atouts qui promettent un beau succès au PC 7000 ; succès déjà constaté lors de sa première présentation sur le stand SHARP SYSTEMES de SICOB.

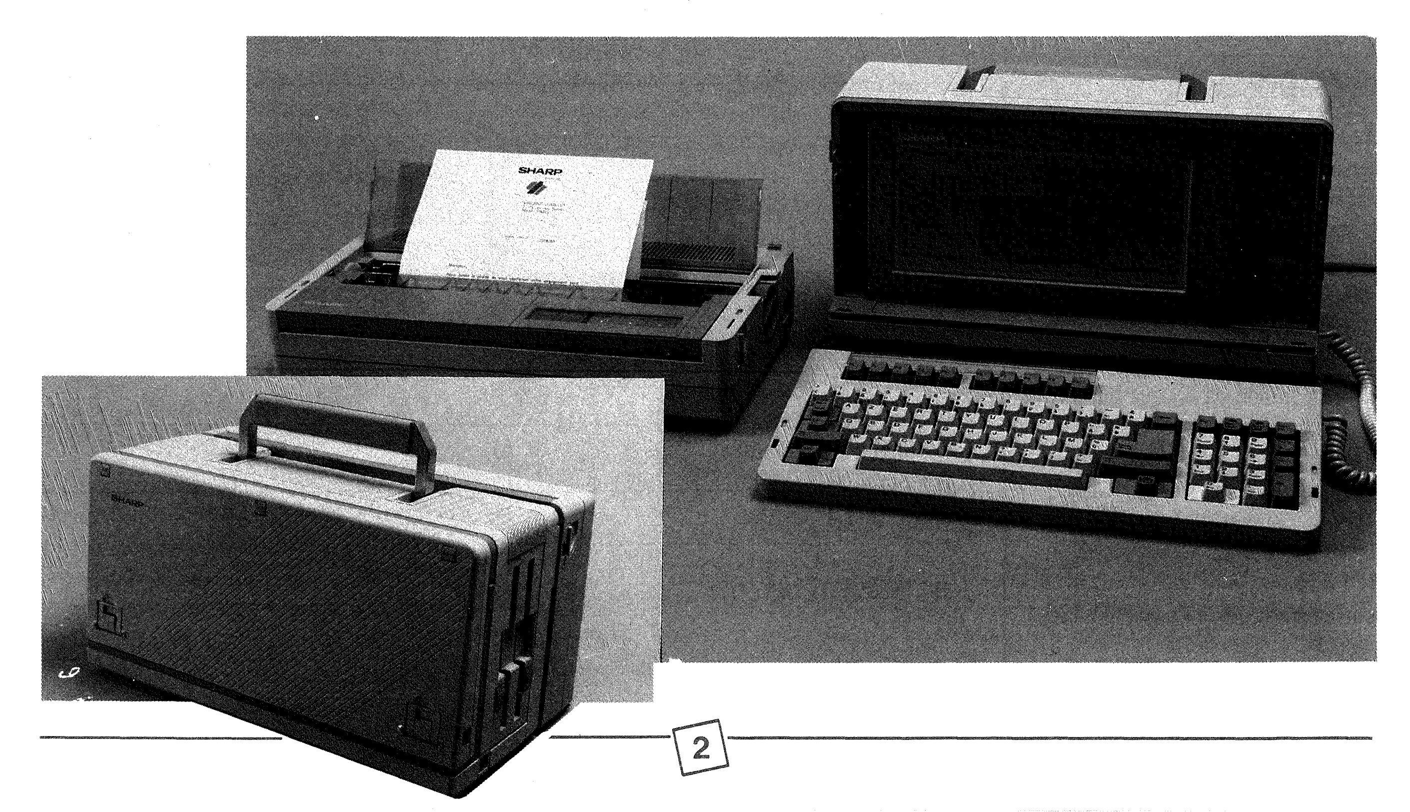
* IBM est une marque déposée par INTERNATIONAL BUSINESS CORPORATION ; MS.DOS par MICROSOFT CORPORATION.

CARACTERISTIQUES TECHNIQUES

Microprocesseurs : I-8086-2 (7.37 MHz), I-8087-2(opt).
Mémoire vive : 384 Ko extensibles à 768 Ko.
Disquettes : 2 unités 5"1/4 de 360/320 Ko formatés.
Ecran : 80 x 25 caractères, 640 x 200 pixels gérés « bit-maped ». Eclairage par panneau électroluminescent avec une fonction d'extinction automatique intégrée.
Clavier : détachable, conforme au PC / AT, 3 leds.
Interfaces : 1 x parallèle CENTRONICS, série RS-232 C.
Dimensions : 410 x 160 x 25 mm.
Poids : 8,5 kg
Alimentation : 220V, 50Hz.

OPTIONS

Adaptateur pour écran couleur:intégrable à l'unité centrale, compatible aux normes IBM*
Boîtier d'extensions : disque dur 3"1/2 de 10 Mo formatés et 3 Slots pour cartes longues ou courtes compatibles IBM*
Transport : bagage souple protégeant l'ordinateur (et si il y a lieu l'imprimante) dans les déplacements.



LES NOUVEAUTÉS

LES NOUVEAUTÉS



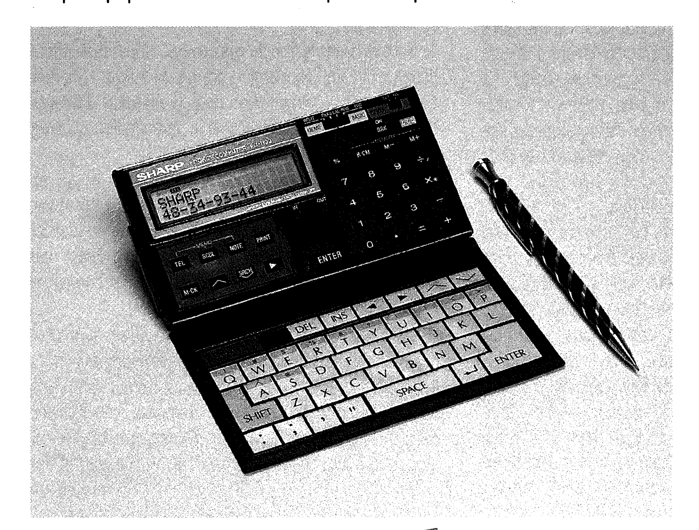
PC 1100 EL 6150

Je dois téléphoner à Monsieur SHARP. Je sors mon carnet d'adresses, trouve le numéro et l'appelle. Au cours de la conversation dont j'écris les grandes lignes sur mon bloc-notes, nous convenons d'un rendez-vous ; je sors mon agenda pour le noter.

Carnet d'adresses, Agenda, Bloc-notes, beaucoup de papiers et de contraintes pour bien peu de notes. Alors... on tire un trait et on découvre les nouveaux carnets de bord électroniques PC 1100 et EL 6150 qui ont justement pour ambition de vous faire jeter au panier vos anciens et encombrants carnets.

Il s'agit, en fait, d'une nouvelle famille de produits réunissant sous un même clavier un ordinateur, une calculatrice, un gestionnaire de fichier et un agenda.

La EL 6150 possède toutes ces fonctions ; équipée de 4 Ko. de mémoire vive, elle permet la mémorisation d'environ 160 numéros de téléphone. Le PC 1100 diffère dans sa structure mémoire. Equipé en version de base d'une carte mémoire de 2 Ko., il peut accepter des cartes de 4 ou 8 Ko. Les informations contenues dans ces mémoires peuvent être protégées des regards indiscrets. Outre les fonctions déjà citées, le PC 1100 est également un ordinateur de poche programmable en BASIC ; BASIC moins riche que celui du PC 1246 car ne possèdant pas de gestion alphanumérique. Imprimante et magnétophone peuvent se connecter sur le PC 1100 puisqu'il est connectable au berceau CE 125.



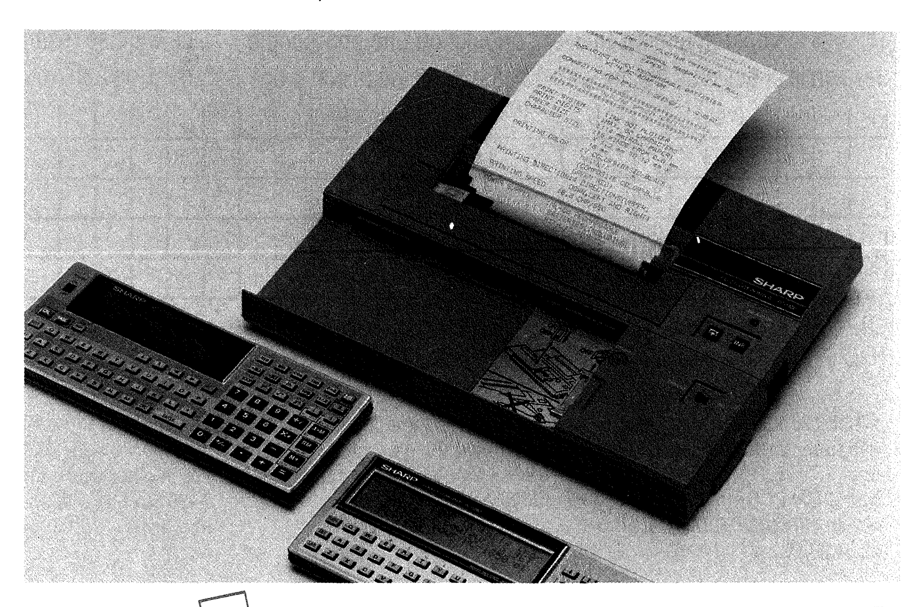


LES NOUVEAUTÉS

CE 140P

Berceau pour les PC 1350 et 1450, le CE 140 P se connecte sur la sortie série de ces 2 produits. C'est une imprimante table traçante d'un nouveau type. Sa principale caractéristique : un jeu de 7 couleurs obtenu par 4 couleurs de base (Noir, Jaune, Magenta et Cyan) et 3 couleurs obtenues par mélange des couleurs de base : Bleu, Vert et Rouge. Le procédé d'impression est un savant dosage de 2 techniques bien connues : le système à aiguilles et le jet d'encre. La tête d'impression n'utilise plus de stylos ou de ruban mais des cartouches d'encre. 4 petites aiguilles (1 par couleur de base) vont déposer l'encre sur le papier. Rapide et propre, le débit d'encre est contrôlé et déposé, non projeté.

Imprimant sur du papier normal de 114 mm en rouleau, le CE 140P se commande comme la table traçante CE 516P ou l'imprimante du PC 2500. Il forme un ensemble compacte, esthétique et autonome avec votre PC 1350 ou 1450. A noter que cette interface ne possède pas de connection pour un magnétophone à cassettes.





BUGS...BUGS...BUGS.

Le BUG, c'était, à l'époque héroïque des ordinateurs à lampes, l'innocent petit moustique qui court-circuitait les électrodes des dites lampes et provoquant le plantage de plusieurs tonnes d'électronique. Depuis, ce terme s'est généralisé à tout défaut provoquant une erreur dans un système informatique.

Des bugs, il en existe également dans le Sharpentier ; heureusement peu nombreux mais toujours très agaçant tant pour vous, lecteur, que pour nous qui faisons le maximum, pour les éviter. Voici ceux que nous avons découvert dans les numéros précédents ; n'hésitez pas à nous contacter si vous en avez découvert d'autres.

N° 12

Page 20 : SOLEIL
dans le programme
Ligne 62 A=ATN (K★TANF)+B
Ligne 145 il manque une parenthèse de fermeture
Ligne 221 elle se termine par −(M>11)
Ligne 241 V=2★ATN (TAN(V/2★√((1+E)/(1−E)))
Ligne 470 IF W > 0
Page 21, colonne de droite, lire A=ATN (SINT/SINL....

N° 14

Page 59, dernière ligne colonne centrale, lire W1, 0010,0100,1200
Page 32, Astuce 2, remplacer les ; par des ,
Page 12 Ne tenez pas compte des valeurs de contrôle de BIDI, elles sont fausses !
Page 13 Il manque :

```
40E0 : BE  41  80  48  44  4A  00  6A  DF
40E8 : 9B  B5  00  BE  45  00  BE  4E  87
```

et remplacer la ligne 5220 par

```
5220 : 89  03  BA  40  E0  B7  45  99  6D
```

Pour lancer le programme faire CALL&40E0 ENTER

LE CLUB ET LES LOGICIELS

Le principe des échanges de logiciels a été créé dès la naissance du Club. Devenu, au fil des années, une véritable institution, il répondait à une de nos vocations premières : Faire profiter à tous des créations et découvertes de chacun. Ces échanges avaient pour but de favoriser une émulation commune aux possesseurs de micros SHARP et de créer un environnement logiciel de qualité accessible à l'ensemble de nos adhérents. La qualité et la quantité de ces logiciels augmentant, nous nous trouvons maintenant confrontés à de très nombreux problèmes liés à la diffusion presque gratuite de ces programmes. Ce service qui devait, à l'origine, répondre à une demande raisonnable de l'ensemble de nos adhérents, s'est maintenant transformé en une véritable industrie. Nous copions et expédions actuellement plus de 200 programmes par semaine, cela représente un travail très long et fastidieux pour les membres bénévoles du Club qui consacrent régulièrement plusieurs heures de leur temps de loisirs à effectuer ces copies ; La limite de 3 programmes par envoi étant rarement respectée ou détournée par l'envoi successif de plusieurs demandes. Ce temps passé au détriment d'un véritable travail de recherches et de contacts constructifs explique, en partie, le retard de parution de ce bulletin.

Nous avons également constaté, ces derniers mois, une importante recrudescence de piratages de nos progammes qui sont, très souvent, légèrement modifiés ou adaptés et revendue à des prix sans aucune commune mesure avec les 15 F. demandés par le club. Ces pratiques lèsent les concepteurs de logiciels qui voient ainsi le fruit de nombreuses heures de travail être utilisées à des fins mercantiles par des intermédiaires peu scrupuleux. Elles discréditent également l'esprit même du Club dont le principal objectif est de diffuser des informations et des logiciels au moindre coût, donc accessibles au plus grand nombre.

Face à ces difficultés, nous avons dû modifier sensiblement le système de distribution des logiciels créés autour des produits micro SHARP. Dorénavant, les programmes reçus au Club seront testés et sélectionnés, certains d'entre eux pourront être commercialisés par le service logiciels du Club, après accord de leurs auteurs. Ces programmes seront également disponibles par l'intermédiaire du réseau de revendeurs SHARP. Comme dans nos précédents bulletins, une logithèque spécifique à chaque machine sera présentée régulièrement ; dans le numéro 16, il s'agira du MZ 800.

Ces différentes modifications ne changent en rien le fonctionnement de notre Club. Vous trouverez, dans ce bulletin, la logithèque de tous les PC autres que PC 1 500. Vous pouvez y accéder, ainsi qu'à toutes les logithèques PC et MZ jusqu'au 31-12-85. Comme par le passé, vous pouvez toujours nous faire parvenir vos propres créations : Astuces, découvertes ou programmes. 3 cas peuvent alors se présenter :

1. Nous publions le fruit de vos travaux. Nous vous offrons, dans ce cas un abonnement gratuit de 1 an au Sharpentier, des points programmes vous permettant d'acquérir des logiciels commercialisés et la participation au tirage au sort du CRITERIUM SHARPENTIERS.

2. Nous envisageons de commmercialiser votre programme. Nous prendrons alors contact avec vous et fixerons ensemble les modalités de diffusion de ce logiciel.

3. Aucun des 2 premiers cas n'est retenu. Nous vous retournerons, dans ce cas votre programme, accompagné, éventuellement, de point(s) programme(s).

Ces décisions devrait mettre un terme à une situation devenue difficilement supportable pour les animateurs bénévoles et permanents du Club. Elle nous permettront également de consacrer encore plus de temps et d'énergie pour résoudre vos problèmes, améliorer nos contacts et à vous proposer un bulletin répondant encore mieux à vos besoins.

Sylvain Bizoirre



Exclusivement réservées aux particuliers pour la vente ou la recherche de matériel d'occasion, votre annonce devra nous parvenir 1 mois avant la parution du bulletin.

Elle devra impérativement utiliser l'original ou une photocopie de la grille ci-dessous (5 lignes de 40 caractères), à raison d'une case par caractère ou espace. Elle mentionnera, en cas de vente, le type exact de l'appareil, la date d'achat, le prix de vente ainsi que vos coordonnées complètes.

Le Club se réserve le droit de refuser toute annonce ne répondant pas aux critères ci-dessus.

Ci-joint le texte de mon annonce à paraitre dans le prochain bulletin, accompagné de son règlement (80 F) à l'ordre du club des SHARPENTIERS.

A SAISIR PC 1 500 (84) 40K, 4 MHZ, + LM, + logiciels : **3 500 F.** module 32 k avec batterie : **1 700 F.** PC 1 500 + CE-155 + logiciels : **1 500 F.** CH. POULIN, 7 bis, rue Mozart, 78800 HOUILLES – tél. : 39.68.93.70.

A VENDRE imprimante CE 150 (06-84) + k7 de programmes. Peu utilisée : **1 300 F.** DENIS RABAULT, 2, rue du Vésinet, 78290 CROISSY.

VDS PC 1 261 (10 K MEV) + manuel + revues du club + CE 124 + progs : **1 590 F.** **VDS** moniteur couleur 36 CM, prise peritel : **2 490 F.** C. COUVIDA. tél. 69.21.29.66 week-end.

VEND PC 1 500 + imprimante + mem. 16 K + magnétophone + cours basic + machine + programmes. Pert-court chemin tb. état ach. (06-83) : **3 700 F.** BILLARD, 173, rue Aristide Briand, 78700 Conflans-ste-Honorine - tél. : 39.19.57.12.



Les affaires club sont essentiellement constituées de matériels, accessoires ou périphériques qui ne sont pas (ou plus) commercialisés par le réseau de revendeurs SHARP. Comme vous pouvez le constater, leurs prix sont très attractifs. Une précision, toutefois, seules, les « Affaires club » sont commercialisées par le club, à l'exclusion de tous autres matériels normalement distribués par les points de ventes SHARP.

SHARP MZ 720

+ BASIC SHARP
+ SUPER BASIC
+ EDIFORTH
+ 10 LOGICIELS DE JEUX

1 290 F

SFD

Unité de disquettes 5 POUCES 1/4 connectable directement sur MZ 700. Capacité de stockage : 320 Koctets formatés. Livrée avec DISC BASIC, programmes utilitaires et manuel.

2 190 F

PC 1500

CE 155	Module mémoire 8 Ko.	250 F
CE 153	Table à digitaliser. Extension de 140 touches sensitives au clavier du PC 1500	550 F
QCN*	Connecteur mâle 60 broches permettant l'accès au bus du microprocesseur	215 F

PC 1251

CE 12 A	Micro-cassette et manuel (en Anglais) traitant de 19 programmes orientés STATISTIQUES	120 F
CE 12 B	Micro-cassette et manuel traitant de 20 programmes orientés CALCULS ELECTRONIQUES	120 F
CE 12 C	Micro-cassette et manuel traitant de 20 programmes orientés MATHEMATIQUES	120 F

PC 1211

CE 122	Interface imprimante	490 F

MZ 80 A

AEU	Panier d'extentions pour cartes disquettes et imprimantes	500 F
AFI	Cartes interface disquettes	600 F
AMD	Master disquette	150 F
ADO2	FDOS avec notice très détaillé pour programmer en assembleur sous disquettes	600 F

MZ 80B

EU	Panier d'extentions pour cartes graphiques imprimantes et disquettes	600 F

MZ 80 K

T41	Langage PASCAL (cassette + notice)	200 F

TOUS MZ 80

MZ 80 FDA1	Lecteur de disquettes 5 pouces 1/4 pour MZ 80 K, A ou B	2 200 F
MZ 80 FDA2	Double lecteur de disquettes 5 p.	3 200 F

BON DE COMMANDE

Je passe commande de :

REF.	QUANT.	DESIGNATION	P.U TTC	P.TOTAL TTC
			TOTAL TTC	

NOM .... PRENOM ....
ADRESSE ....

Ci-joint mon règlement de .... F, par chèque bancaire ou C.C.P. établi à l'ordre du SHARP BUROTYPE MACHINES*. Il représente le montant total TTC de ma commande. Je prends bonne note qu'au cas ou ma commande ne pourrait être honorée dans la limite des stocks disponibles, je serai intégralement remboursé du montant des articles non livrés.

DATE .... SIGNATURE

* Pour le connecteur QCN, règlement à l'ordre du « Club des Sharpentiers ».

DANS VOTRE BIBLIOTHEQUE...

PC 1350 MACHINE LANGAGE REFERENCE MANUAL

Comme ses prédécesseurs, le PC 1350 possède maintenant son manuel de référence. Il explique et commente le fonctionnement du microprocesseur SC 61860, et son implantation dans le PC 1350 (Pointeurs, Memory Map, adresse système). Une description HARD du PC et de ses périphériques complète ce manuel indispensable pour qui veut développer ses connaissances sur ce produit. Cet ouvrage (en anglais) peut également être utilisé par les possesseurs de PC 2500 dont le microprocesseur, donc toute la partie LM est rigoureusement identique.

MANUELS L.M PC 1500 PC 1251 MANUAL MZ 800



M. ....
ADRESSE ....
....
vous prie de bien vouloir lui faire parvenir
☐ Recueil(s) des n° 1 à 5 des bulletins SHARPENTIERS au prix de 110 F. (port compris)
☐ Bulletin(s) SHARPENTIER n° 6 ☐, 7 ☐, 8 ☐, 9 ☐, 10 ☐ au prix de 20 F chacun
☐ Bulletin(s) SHARPENTIER n° 11 ☐, 12 ☐, 13 ☐ au prix de 25 F chacun. n° 14 ☐ au prix de 30 F.
☐ Manuel(s) Langage Machine PC 1500 ☐, PC 1251 ☐, PC 1350 ☐ au prix de 180 F chacun.
☐ Manuel de référence MZ800 au prix de 240 F l'un

Signature

Ci-joint un chèque de.......... francs.
Date

Ces 4 manuels, en Anglais, très bien documentés, sont des ouvrages de référence et n'abordent pas l'initiation au langage machine. Ils sont disponibles directement au Club, le mercredi après-midi (exclusivement), au prix de 170 F (MZ800 : 230 F) l'un ; ou par correspondance au prix de 180 F (MZ800 : 240 F) (port compris). Le manuel LM PC 1251 s'applique également à la série PC 14XX, aux PC 1260.

SHARP ENSEIGNEMENT

Lors de la journée Educatique 85, il avait été proposé la création de planches pédagogiques destinées à démontrer, par l'image, les différentes étapes du fonctionnement et de l'utilisation d'un micro-ordinateur. Notre première planche est, dès maintenant disponible. Elle détaille et explique clairement toutes les instructions du langage BASIC exécutables sur un ordinateur de poche. Au format 140 ★ 100 cm, elle peut aisément être affichée en salle de cours.

Si vous êtes enseignant, vous pouvez vous procurer cette planche chez votre Point de Vente Enseignement ou au département Calcul de la Sté. ; S.B.M.

Nous envisageons la création de différentes autres planches de ce type. Pour que celles-ci répondent au mieux à vos besoins, nous aimerions que vous nous exprimiez vos souhaits et vos suggestions.

Vos avis détermineront l'intérêt pédagogique et l'apport pratique que peuvent apporter ces planches au sein de vos cours. N'hésitez pas, pour cela, à contacter le Club ou le service PROMOTION de la Sté. S.B.M.

Sté. S.B.M. Département Calcul ou Promotion 151-153, av. J. Jaurès
93307 Aubervilliers cedex

Simulation des expériences aléatoires sur ordinateur

Considérons un dé parfaitement équilibré et lancé loyalement. A chaque lancer, le chiffre obtenu est purement aléatoire. On pourra donc, en lançant le dé un grand nombre de fois et en notant le chiffre obtenu, construire une « table de nombres aléatoires » entiers compris entre 1 et 6.

Cette expérience pourrait être effectuée avec un pièce de monnaie, une roulette de casino ... ou tout autre objet, ce qui permettrait de construire d'autres tables de nombres aléatoires.

1. GENERALITES

Il est cependant évident que procéder ainsi est assez pénible. Aussi a-t-on cherché des procédés permettant d'engendrer des suites de nombres ayant pratiquement les propriétés des suites aléatoires. De telles suites sont nommées ; pseudo-aléatoires. A partir d'un « nombe source » x_0 (on dit parfois la « semence ») un certain procédé (dit « générateur aléatoire ») permet de construire d'autres nombres qui peuvent être considérés comme aléatoires. En fait, si on recommence plusieurs fois le même processus à partir du même nombre source x_0, on engendrera toujours la même suite de nombres (c'est en cela que la suite n'est pas vraiment aléatoire), mais de nombreux tests statistiques permettent de montrer que la suite de nombres pseudo-aléatoires est très voisine d'une véritable suite aléatoire.
De nombeux générateurs ont passé avec succès ces tests statistiques et sont utilisés abondamment par les ordinateurs pour effectuer très rapidement des simulations d'expériences aléatoires qui demanderaient des dizaines ou peut-être des centaines d'années de calcul ou d'expérimentation si on voulait les réaliser effectivement.

Dans la suite, nous utiliserons le générateur incorporé à notre ordinateur, en supposant qu'il nous donne des nombres compris entre 0 et I (c'est le cas de RND(I) pour les ordinateurs de « table » ou de RND(O) pour le PC-1500). Sur une calculatrice programmable, ne possédant pas ce générateur automatique, on pourra utiliser l'un ou l'autre des deux algorythmes suivants :

Le 147-générateur :
On se donne x_0 compris entre 0 et I (par exemple 0,153938) et on construit la suite x_n par : x_{n+1} = partie fractionnaire de $(147.x_n)$

Le π -générateur :
On choisit x_0 comme précédemment et on définit x_{n+1} par x_{n+1} = partie fractionnaire de $(x_n+\pi)^5$

Pour que les suites engendrées soient convenables, il faudra autant que possible, prendre un x_0 avec beaucoup de décimales, les plus variées possibles !

2. QUELQUES SIMULATIONS SIMPLES :

Lancer d'une pièce de monnaie :
On conviendra de représenter Pile par 0 et face par I. Il nous faut donc engendrer une suite aléatoire de 0 et de I. Comme les x_n sont dans (0 – 1), nous remarquons que $2.x_n$ est dans (0 – 2) donc sa partie entière sera soit 0 soit I.
Le programme suivant permet l'affichage de la suite aléatoire engendrée. Sur l'ordinateur de poche, on fera PAUSE au lieu de PRINT et un BEEP matérialisera le changement de x_n

PC-I500

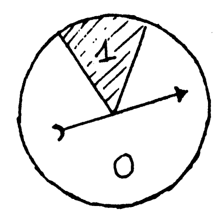
```
10 X=INT(2*RND(1))      10 X=INT(2*RND(0)):BEEP 1
20 PRINT X              20 PAUSE X
30 GOTO 10              30 GOTO 10
```

Lancer d'un dé :
On remarquera que la partie entière de $6.x_n$ prend les valeurs 0,I, ...,5. Il suffira donc de calculer I+INT(6.xn) pour avoir un générateur donnant I,...,6

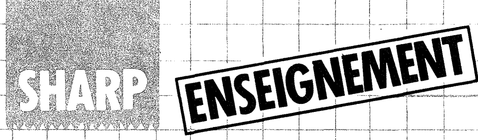
```
10 X=1+INT(6*RND(1))  10 X=1+INT(6*RND(0)):BEEP1
20 PRINT X            20 PAUSE X
30 GOTO 10            30 GOTO 10
```

Roulette :
Considérons une roulette pafaite divisée en deux secteurs marqués 0 et 1, la surface « 0 » étant les 7/10e de la surface totale. L'aiguille va donc s'arrêter sur le 0 avec la probabilité 0,3. La simulation de cette expérience se fera en affichant la partie entière de x_n + 0,3

```
10 X=INT(RND(1)+.3)   10 X=INT(RND(0)+.3):BEEP 1
20 PRINT X            20 PAUSE X
30 GOTO 10            30 GOTO 10
```



Simulation d'un déplacement aléatoire sur un axe :
Une particule partant de 0 se déplace sur l'axe réel. A chaque fois, elle ne peut se déplacer que d'une unité avec la probabilité .5 pour chaque direction. On désire que l'ordinateur affiche l'abscisse de la particule.



Remarquons que $2.INT(2.x_n) - 1$ soit -1 de façon équiprobable. Il nous suffira d'ajouter ce nombre à l'abscisse de la particule, pour voir se réaliser son déplacement aléatoire :

```
10 X=0                        10 X=0
20 X=X+2*INT(2*RND(1))-1      20 X=X+2*INT(2*RND(0))-1
30 PRINT X                    30 PAUSE X
40 GOTO 20                    40 GOTO 20
```

Remarque :
Si la particule devait aller vers les x positifs avec la probabilité 0.6 et donc 0.4 vers les x. négatifs, il suffira de remplacer RND(1) par RND(1) + .6

3. ESTIMATION D'UNE ESPERANCE :

Nous utilisons maintenant le générateur aléatoire pour simuler un grand nombre d'expériences consistant à lancer un dé. Nous appelons Xi le point obtenu au i[e] lancer et nous calculons $(S_1 + X_2 + ... + X_n)/n$, moyenne des points obtenus lors des n premiers lancers.
La loi des grands nombres nous dit que S_n doit converger en probabilité vers l'espérance commune des X_i c'est-à-dire 7/2.

Lorsque ce programme tourne, on doit voir apparaître à l'affichage, des nombres oscillant autour de 3.5.

```
10 N=0:S=0
20 X=1+INT(6*RND(1))
30 N=N+1:S=S+X
40 PRINT S/N
50 GOTO 20
```

Cet exemple illustre bien la notion de convergence en probabilité : lorsque n est grand, il y a de très fortes chances pour que le nombre affiché soit voisin de 7/2, mais cela ne prouve pas qu'il est obligatoirement voisin de cette valeur. Cela signifie que si on programmait de façon identique, un très grand nombre de machines, alors, lorsque n devient grand, seul un petit nombe de ces machines afficherait des nombres très différents de 7/2.

4. ESTIMATION D'UNE PROBABILITE :

Les générateurs aléatoires permettent de se faire une idée de la probabilité d'un événement, lorsqu'elle est difficilement calculable. Pour le montrer, nous prenons un exemple très simple, celui du lancer d'un dé et nous essayons de retrouver 1/6 = 0.1666666... comme probabilité d'apparition du chiffre 6. Le programme suivant permet d'afficher la fréquence du nombre d'apparitions du 6 au cours des n premières épreuves :

Dans ce programme, N compte le nombre d'épreuves effectuées et S le nombre de 6 obtenus.

```
10 N=0:S=0
20 X=1+INT(6*RND(1))
   :N=N+1
30 IF X=6 LET S=S+1
40 PRINT S/N
50 GOTO 20
```

Nous savons, d'après la loi des grands nombres (*) que cette fréquence converge en probabilité vers **1**/6 et c'est ce que l'on constate à l'affichage de la machine.

* Une épreuve E conduit à l'apparition de A (probabilité P) ou Ā (probabilité $q = \mathbf{1} - p$). On répète E, un grand nombre de fois (de façons indépendantes) on note $X_i = \mathbf{1}$ si A se produit à l'épreuve n° i et $X_i = 0$ si non. Soit S la fréquence d'apparition de A, c'est-à-dire $S_n = (X_1 + X_2 + ... + X_n)/n$.
On sait, d'après la loi des grands nombres, que S_n va converger en probabilité vers l'espérance commune des X_i c'est-à-dire $0.(\mathbf{1} - p) + \mathbf{1}.p = p$, probabilité d'apparition de A.

Autre exemple :
celui de la roulette du § 2
Essayons de retrouver la probabilité 0.3 d'obtenir le « **1** » en faisant tourner la roulette :

Ici, S/N doit osciller autour de 0.3

```
10 N=0:S=0
20 X=INT(RND(1)+.3):N=N+1
30 IF X=1 LET S=S+1
40 PRINT S/N
50 GOTO 20
```

5. CALCUL D'INTEGRALES PAR LA METHODE DE MONTE-CARLO

Soit X une variable aléatoire continue sur (a,b) de densité $g(x) = 1/(b-a)$ sur cet intervalle (et 0 en dehors). Soit f définie continue sur (a,b) à valeurs dans R. Notons $Y = f(X)$, la fonction de la variable X. On sait que l'espérance de Y est donnée par :

$$E(Y) = \int_a^b f(x)g(x)dx \text{ d'où: } \int_a^b f(x)dx = (b-a)E(Y)$$

Effectuons une série d'expériences aléatoires indépendantes, chacune d'elles consistant à choisir, au hasard, un point dans (a,b). A la i[e] expérience, notons X_i l'abscisse du point obtenu. Il est facile de voir (cela a été fait en cours pour l'intervalle (0.1)) que X_i suit la loi de X, c'est-à-dire la loi continue uniforme sur (a,b). Par conséquent, les variables $Y_i = f(X_i)$ vont être indépendantes et suivre la loi de Y. La loi des grands nombres nous dit que, dans ces conditions, $(Y_1 + Y_2 + ... + Y_n)/n$ va converger en probabilité vers E(Y).
On voit ainsi apparaître un moyen de calculer l'intégrale $\int_a^b f(x)dx$ par simulation :

On contruit une suite x'_n uniformément répartie dans (a,b). Pour cela, nous reprenons nos x_n qui se trouvaient dans (0.1) et nos posons : $x'_n = (b-a)x_n + a$.
Comme $(f(x'_1) + f(x'_n))/n$ converge (en probabilité) vers E(Y),

$$\frac{b-a)}{n}(f(x'_1) + ... + f(x'_n)) \text{ doit converger vers } \int_a^b f(x)dx$$

Voici quelques exemples de programmes où les bornes A et B sont 0 et 1 (ligne 10) et où la fonction $y = f(x)$ est écrite à la ligne 70. A côté de chaque programme, on trouvera la valeur « exacte » de l'intégrale.

$f(x) = x^3$ entre 0 et 1 : 0.25

```
10 N=0:S=0:A=0:B=1
20 X=A+(B-A)*RND(1):GOSUB 70
30 S=S+Y:N=N+1
40 PRINT S/N
50 GOTO 20
60 REM ~~~~~~~~~~~~~
70 Y=X↑3
80 RETURN
```



$f(x) = 1/(1 + x^2)$ entre 0 et 1 : $\pi/4$
soit .785 398 163 4

```
10 N=0:S=0:A=0:B=1
20 X=A+(B-A)*RND(1):
   GOSUB 70
30 S=S+Y:N=N+1
40 PRINT S/N
50 GOTO 20
60 REM ~~~~~~~~~~~~
70 Y=1/(1+X↑2)
80 RETURN
```

$f(x) = e^{-x^2/E}./\sqrt{E2II}$ entre 0 et 1
0.341 344 746 1

```
10 N=0:S=0:A=0:B=1
20 X=A+(B-A)*RND(1)
   :GOSUB 70
30 S=S+Y:N=N+1
40 PRINT S/N
50 GOTO 20
60 REM ~~~~~~~~~~~~
70 Y=EXP(-X*X/2)/SQR(2*π)
80 RETURN
```

$f(x) = \sin(x)/(1 + x^2)$ entre 0 et 1 :
0.321 793 544 8

```
10 N=0:S=0:A=0:B=1
20 X=A+(B-A)*RND(1)
   :GOSUB 70
30 S=S+Y:N=N+1
40 PRINT S/N
50 GOTO 20
60 REM ~~~~~~~~~~~~
70 Y=SIN(X)/(1+X↑2)
80 RETURN
```

Calcul des déterminants

PC-1261

Ce programme sans prétention permet de calculer la valeur de déterminants numériques d'ordre 2 à 6 compris. Comme il est écrit en BASIC, il ne faut pas être trop pressé dans le cas des déterminants d'ordre 5 et 6.

Il est possible d'afficher le déterminant sur l'imprimante, à condition que les termes ne contiennent qu'un nombre réduit de signes et de chiffres.

On peut remarquer que le calcul d'un déterminant d'ordre 6 (lignes 499 à 550) n'a pu être conduit suivant le même algorithme que les déterminants d'ordre 3,4,5 (lignes 199 à 470). En effet, le PC 1261 n'accepte que cinq niveaux pour les boucles FOR... NEXT.

Daniel MAGNIN

DETERMINANTS

```
  1:REM Daniel MAGNIN 19
    85
  9:REM initialisation
 10:CLEAR : CURSOR 29:
    PRINT "Determinants"
    : INPUT "avec imprim
    ante ? ";FF$
 20:"A" INPUT "ordre de
    2 a 6 ? ";N
 30:N= INT N: IF N<2 OR
    N>6 THEN "A"
 40:DIM A(N,N)
 50:FOR I=1 TO N: FOR J=
    1 TO N
 60:WAIT 0: CLS : PRINT
    "A(";I;J;") ";:
    INPUT A(I,J)
 70:NEXT J: NEXT I
 80:IF LEFT$ (FF$,1)="O"
    GOSUB 600
 85:CLS : GOSUB 950
 90:ON N-1 GOTO 100,200,
    300,400,500
 99:REM ordre 2
100:GOSUB 150:DE=S: GOTO
    900
149:REM ordre 2: calcul
150:S=A(1,1)*A(2,2)-A(1,
    2)*A(2,1): RETURN
199:REM ordre 3: mise en
     place
200:GOSUB 850
210:FOR I=1 TO 3: FOR J=
    1 TO 3:B(I,J)=A(I,J)
    : NEXT J: NEXT I
220:GOSUB 250:DE=T: GOTO
    900
249:REM calcul: ordre 3
250:FOR H=1 TO 3:S=0:P=3
    :Y=H: GOSUB 700
260:GOSUB 150:T=T+S*B(H,
    1)*(-1)^(H+1): NEXT
    H: RETURN
299:REM ordre 4: mise en
     place
300:GOSUB 840
310:FOR I=1 TO 4: FOR J=
    1 TO 4:C(I,J)=A(I,J)
    : NEXT J: NEXT I
320:GOSUB 350:DE=U: GOTO
    900
349:REM calcul ordre 4
350:FOR K=1 TO 4:T=0:P=4
    :Y=K: GOSUB 700
360:GOSUB 250:U=U+T*C(K,
    1)*(-1)^(K+1): NEXT
    K: RETURN
```

```
399:REM ordre 5: mise en
     place
400:GOSUB 830
410:FOR I=1 TO 5: FOR J=
     1 TO 5:D(I,J)=A(I,J)
     : NEXT J: NEXT I
420:GOSUB 450:DE=V: GOTO
     900
449:REM calcul ordre 5
450:FOR L=1 TO 5:U=0:Y=L
     :P=5: GOSUB 700
460:GOSUB 350:V=V+U*D(L,
     1)*(-1)^(L+1): NEXT
     L: RETURN
470:GOSUB 330:V=V+U*D(E,
     1): NEXT E: RETURN
499:REM ordre 6
500:GOSUB 820
510:FOR I=1 TO 6: FOR J=
     1 TO 6:E(I,J)=A(I,J)
     : NEXT J: NEXT I
520:M=M+1:V=0:Y=M:P=6:
     GOSUB 700
530:GOSUB 450:W=W+V*E(M,
     1)*(-1)^(M+1)
540:IF M<6 THEN 520
550:DE=W: GOTO 900
599:REM impression du de
     terminant
600:PRINT = LPRINT :
     PRINT "determinant "
     ; STR$ N;"*"; STR$ N
610:FOR I=1 TO N: FOR J=
     1 TO N
620:Q= LEN STR$ A(I,J):
     IF J=1 OR N>5 LET Q=
     Q+2
630:ON Q GOSUB 650,660,6
     70,680
640:PRINT STR$ A(I,J);:
     NEXT J: PRINT "":
     NEXT I: RETURN
650:PRINT "    ";:
     RETURN
660:PRINT "   ";: RETURN
670:PRINT "  ";: RETURN
680:PRINT " ";: RETURN
699:REM formation des mi
     neurs
700:FOR I=1 TO P:Z=I: IF
     I>=Y LET Z=I+1
710:IF Z>P THEN 730
720:FOR J=2 TO P: ON P-2
     GOSUB 750,760,770,78
     0: NEXT J
730:NEXT I: RETURN
750:A(I,J-1)=B(Z,J):
     RETURN
760:B(I,J-1)=C(Z,J):
     RETURN
770:C(I,J-1)=D(Z,J):
     RETURN
780:D(I,J-1)=E(Z,J):
     RETURN
819:REM dimensionnement
     des memoires
820:DIM E(6,6)
830:DIM D(5,5)
840:DIM C(4,4)
850:DIM B(3,3): RETURN
899:REM affichage du res
     ultat
900:BEEP 1: CLS : WAIT :
     PRINT "delta "; STR$
     N;"="; STR$ DE
910:PRINT = PRINT : GOTO
     10
949:REM correction des e
     rreurs d entree
950:INPUT "des modificat
     ions ? ";FE$
960:IF LEFT$ (FE$,1)<>"O
     " RETURN
970:INPUT "ligne ";I,"co
     lonne ";J,"valeur ";
     A(I,J)
980:IF LEFT$ (FF$,1)="O"
     PRINT "A(";I;J;")=";
     A(I,J)
990:GOTO 950
```

Sharp "enseignants" Classements

Vous êtes-vous trouvés, chers collègues instituteurs et professeurs, un soir de rentrée, en face de piles de fiches remplies par les élèves ? Sûrement, et vous vous êtes alors mis en devoir de classer tous ces noms par ordre alphabétique, afin d'établir une liste de votre ou de vos classes. C'est là un travail du plus haut intérêt intellectuel, de plus plein d'imprévu quand par inadvertance vous avez inversé deux noms. Le petit logiciel « CLALI » vous facilite cette corvée. Vous entrez les noms dans l'ordre où ils se présentent, et à condition de ne pas être pressé, la machine vous livrera la liste de vos disciples par ordre alphabétique. Si vous le désirez, elle pourra l'imprimer, et même la sauvegarder sur une cassette. Vous n'avez même pas besoin de connaître exactement l'effectif : il vous suffit de l'évaluer grosso-modo et d'introduire cette évaluation en la majorant.

Si dans quelques mois vous avez à nouveau besoin d'imprimer cette liste, le logiciel « LISTI » rappellera votre liste depuis la cassette et vous la reproduira.

Tout cela est bel et bon, mais pas très réjouissant : mieux vaut penser à la fin du trimestre. Il existe encore bien des sections où les élèves sont notés, puis classés par ordre de mérite suivant leur moyenne. Encore un travail exaltant ! Mais le logiciel « CMXI » peut nous venir en aide. Il suffit d'entrer le nom de chaque élève, son nombre de notes et ces notes elles-même. Si la liste alphabétique des élèves de la classe a été enregistrée sur cassette, on n'aura même pas à introduire les noms, le magnétophone s'en chargera.

La moyenne générale de la section est rapidement obtenue. Après quoi, l'opérateur aura le temps de lire son journal, car le logiciel est très lent — d'autant plus lent que la classe est plus surchargée.

Cette lecture terminée, la machine sera vraisemblablement en mesure de fournir — et éventuellement elle aura déjà imprimé — ce classement par ordre de mérite. Je vous laisse le plaisir de découvrir ce qui arrive lorsque l'on a des ex-aequo...

Daniel MAGNIN

CLALI

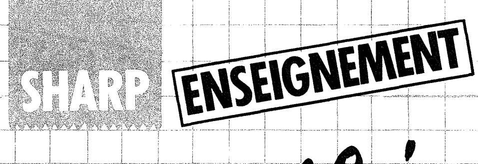
```
  1:REM Daniel MAGNIN 19
    83 version 1985 pour
    PC-1261
  4:REM initialisation
  5:CLEAR : PRINT "Class
    ement alphabetique":
    INPUT "Nombre d elev
    es connu ? ";Z$
 10:INPUT "impression de
     liste ? ";X$
 15:IF LEFT$ (Z$,1)="N"
    THEN 99
 20:INPUT "combien ? ";N
    : DIM AA$(N)
 24:REM entree des noms
    (nombre connu)
 25:FOR I=1 TO N: INPUT
    "nom ? ";AA$(I):
    NEXT I
 29:REM tri
 30:FOR K=1 TO N-1: IF A
    A$(K)<AA$(K+1) THEN
    40
 35:XX$=AA$(K):AA$(K)=AA
    $(K+1):AA$(K+1)=XX$:
    J=1
 40:NEXT K
 45:IF J=1 LET J=0: GOTO
    30
 49:REM impression event
    uelle de la liste
 50:IF LEFT$ (X$,1)="O"
    PRINT = LPRINT
 55:FOR I=1 TO N: PRINT
    I;")";AA$(I): NEXT I
 59:REM sauvegarde optio
    nnelle sur cassette
 60:PRINT = PRINT :
    INPUT "sauvegarde ?
    ";W$: IF LEFT$ (W$,1
    )="N" THEN 85
 65:PRINT #N: PRINT #AA$
    (*)
 85:PRINT "FIN": END
 98:REM entree des noms:
     nombre inconnu
 99:INPUT "donnez un nom
    bre majore ";Y: DIM
    AA$(Y)
110:BEEP 1: PRINT "Apres
     la fin de la liste
    repondez par 0"
120:N=N+1: INPUT "nom ?
    ";AA$(N)
130:IF AA$(N)<>"0" THEN
    120
140:N=N-1: GOTO 30
```

LISTI

```
200:CLEAR : PRINT "Liste
    ": INPUT "impression
     ? ";Z$: IF LEFT$ (Z
    $,1)="O" PRINT =
    LPRINT
210:INPUT #N: DIM AA$(N)
    : INPUT #AA$(*):
    BEEP 1
220:FOR I=1 TO N: PRINT
    I;")";AA$(I): NEXT I
230:PRINT = PRINT :
    PRINT "FIN": END
```

CMXI

```
1000:REM Daniel MAGNIN
     1983 version 1985
     pour PC-1261
1009:REM initialisation
1010:"M" CLEAR : PRINT
     "Moyennes et class
     ement"
1020:WAIT 0: INPUT "ave
     c imprimante ? ";F
     E$
1030:INPUT "liste sur c
     assette ? ";FF$
1039:REM liste manuelle
1040:IF LEFT$ (FF$,1)="
     O" THEN 1500
1050:PRINT "nombre d";
     CHR$ 39;"eleves ";
     : INPUT N: GOSUB 1
     450
1059:REM entree des not
     es
1060:FOR I=1 TO N: CLS
     : PRINT "eleve ";
     STR$ I;" ";: INPUT
     AA$(I)
1070:GOSUB 1600: NEXT I
1099:REM calcul de la m
     oyenne
1100:C=D/N:P=4: IF C<10
     LET P=3
1105:CLS : WAIT : IF
     LEFT$ (FE$,1)="O"
     PRINT = LPRINT :
     WAIT 0
1110:GOSUB 1900: BEEP 1
     : PRINT "moyenne g
     enerale "; STR$ C:
     PRINT = PRINT
1119:REM classement
1120:FOR T=1 TO N-1
1130:IF B(T)>=B(T+1)
     THEN 1150
1140:X=B(T+1):B(T+1)=B(
     T):B(T)=X:J=1
1150:NEXT T
1160:IF J=1 LET J=0:
     GOTO 1120
1199:REM resultats
1200:IF LEFT$ (FE$,1)="
     O" PRINT = LPRINT
1210:BEEP 2: FOR I=1 TO
     N: FOR J=1 TO N
1220:IF C(J)=B(I) GOSUB
     1300
1230:NEXT J: NEXT I:
     PRINT "Fin": END
1299:REM impression du
     classement
1300:P=3:C=B(I): IF C<1
     0 LET P=2
1310:GOSUB 1900: PRINT
     I;")";AA$(J);" ";
     STR$ C: RETURN
1449:REM dimensionnemen
     t
1450:DIM AA$(N): DIM B(
     N): DIM C(N):
     RETURN
1499:REM entrees par ca
     ssete
1500:WAIT : PRINT "prep
     arez le magnetopho
     ne"
1510:INPUT #N: GOSUB 14
     50: INPUT #AA$(*)
1520:FOR I=1 TO N: CLS
     : WAIT : PRINT AA$
     (I): GOSUB 1600
1530:NEXT I: GOTO 1100
1599:REM entree des not
     es
1600:WAIT 0: INPUT "com
     bien de notes ? ";
     Z:V=0
1610:FOR J=1 TO Z: CLS
     : PRINT "note ";J;
     " ";: INPUT X
1620:V=V+X: NEXT J
1630:CLS : PRINT "corre
     ct pour ";AA$(I);"
      ";: INPUT FD$
1640:IF LEFT$ (FD$,1)="
     N" WAIT : CLS :
     PRINT "recommencez
      ";AA$(I);" !":
     GOTO 1600
1650:D=D+V/Z:B(I)=V/Z:C
     (I)=V/Z: RETURN
1899:REM arrondis
1900:A=C:B=P- INT ( LOG
     ( ABS (C)))
1910:C= INT (C*10^B+.5)
     *10^-B
1920:RETURN
```



Les fonctions mathématiques du Basic

Je ne tiens pas à faire de vous de brillants mathématiciens (pardon si vous l'êtes !) mais vous trouverez dans cet exposé quelques notions fondamentales sur les fonctions mathématiques.
Pas de tableaux de variation, de courbes, de limites ou de dérivés, juste un cercle trigonométrique qui traîne avec sinus et cosinus, et surtout des astuces de calculs. Cette étude, écrite pour le PC-1500 peut très bien s'adapter aux basics des différents PC et MZ.

LE MODE ANGULAIRE

Le PC-1500 travaille dans les trois modes angulaires courants que l'homme a inventés. Il aurait été bien sûr trop simple de faire un tour de 0 à 100, c'est pourquoi on a :

DEGREE	1 tour = 360	le + commun
RADIAN	1 tour = 2π	le + logique
GRAD	1 tour = 400	inutilisé

Une manière de convertir un angle d'un système à un autre est :

$$\text{angle système}_2 = \text{angle système}_1 \times \left(\frac{\text{tour système 2}}{\text{tour système 1}}\right)$$

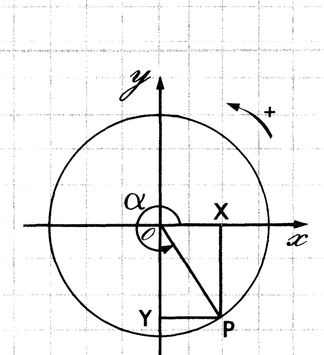
exemple : 90 degrés en gradian $= 90 \times \left(\frac{400}{360}\right) = 100$ gradians

L'apparition du ↑ dans la définition d'un cercle en radian vous rappelle certainement cette formule :
circonférence = $2 \times \pi \times$ rayon
Vous constatez que si l'on prend rayon = 1, on a :
circonférence = $2 \times \pi$ ce qui correspond à son angle en radians !
Ce n'était donc pas si idiot que cela de prendre 1 tour = 6,283185307 radians. Ce cercle de rayon = 1 est souvent utilisé car on n'a plus à trainer R qui n'est qu'un facteur de proportionnalité. Son nom est cercle trigonométrique.
Vous connaissez ? Et bien tant pis ! Continuons... Sinus et Cosinus (dans une nouvelle aventure d'Astérix ?).

Pour calculer la valeur d'un angle sur ce cercle, on a défini une origine, l'axe ox et un sens positif, inverse de celui des aiguilles d'une montre.

Ce cercle est représenté ainsi :

Choisissons un point P quelconque sur le cercle. Sa distance avec le centre est 1 (sinon ce n'est pas un cercle !). On peut calculer avec un rapporteur l'angle formé par OP et Ox. On le nomme x. De même, on peut projeter l'image de ce point sur Ox, on aura alors la distance OX et sur Oy, on aura cette fois Oy.

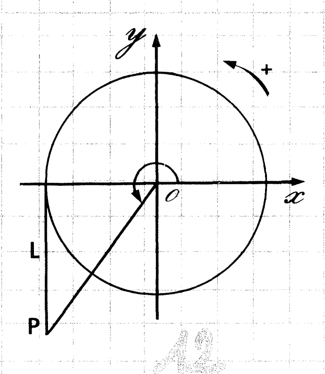


On s'aperçoit alors que OX = COS x OY = SIN x
Pour un cercle de rayon R, il suffit de multiplier par R.
On a donc l'équation d'un cercle ayant son centre en : (0,0).
(X,Y) = (R*COSx, R*SINx)x variant de 0 à 2 π radians.
On s'aperçoit que : $x + 2K\pi = x$ avec K entier
On a des quantités de formule avec ces SIN et COS, voici la plus importante :
$1 = SIN^2x + COS^2x$

TANGEANTE

Cette fonction est en fait : $L = TANx = \frac{SINx}{COSx}$

Remarquez que si $x = \frac{\pi}{2} + K\pi$ avec K entier
COSx = 0 donc TANx n'est pas défini.
Voici ce que cette fonction exprime :



Pour un cercle de rayon R, on aura : L = R*TANx
On définit aussi la fonction contangente qui est :
$\frac{COSx}{SINx} = \frac{1}{TANx}$
La distance L calculée ne sera plus verticale mais horizontale.

ASN, ACS, ATN

Permettent de connaître un angle connaissant une distance. Le paramètre des fonctions n'est pas influencé par le mode angulaire, mais le résultat l'est. On peut donc convertir 90° en gradians ainsi :
DEGREE : A=SIN 90 : GRAD : A=ASN A
Les paramètres de ASN et ACS seront toujours compris entre −1 et 2 car
−1≤COSx≤1 et
−1≤SINx≤1

SQR ET ↑

SQR sert à calculer une racine carrée. Notez qu'il est plus économique d'utiliser le signe √ qui n'utilise qu'un octet au lieu de deux pour SQR
a↑b calcule la fonction a puissance b.
Cette fonction est très lente, et il est souvent beaucoup plus rapide de faire directement l'opération, si c'est bien sur faisable. Par exemple, on remplacera a↑3 par a*a*a.
Ces deux fonctions introduisent des erreurs de calcul sur le douzième chiffre significatif, et, si vous êtes sûr d'avoir un résulat sans partie fractionnelle, faites, si R est le résultat :
INT (R+.1)
pour être sûr d'avoir le bon résultat.
La racine Nième d'un nombre s'obtient par :
a↑(1/N)
par exemple :
racine cubique de 8
8↑(1/3) ---> 2
et SQR a=a (1/2)

LOGARYTHME DECIMAL

Cette fonction LOG est très utile car elle permet de savoir où est le premier chiffre significatif, différent de 0, d'un nombre. Soit à ce nombre, INT LOG a vous donnera la place relative du premier chiffre significatif par rapport au chiffre de l'unité.
a=A B C D. E F G H
INTLOG a= 3 2 1 0 −1 −2 −3 −4
exemple :
a=A B C D. E F G H
=0 0 3 1. 4 1 0 0
INTLOGa=1

Calculer un logarythme de 0 donne une ERREUR 39 car il n'y a pas de chiffre significatif.

LOGARYTHME NEPERIEN ET EXPONENTIELLE

$LNa = \frac{LOGa}{LOGe}$ $EXPa = e^a$
avec e = EXP1 = 2,718...
Pas d'utilisation pratique trouvée autre que mathématique.

SGN, ABS

Ces fonctions sont assez simples :
SGNa renvoie le signe de a
a<0 --- −1
a=0 --- 0
A > 0 --- 1
ABSa renvoie la valeur absolue de a = a
ABSa=√Eaxa

INT

INT 3,432=3
INT 0,001=0
INT 3421,011=3421
mais attention
INT −2=−2
INT −2,1 =−3
INT n'est donc pas exactement l'équivalent du ENT mathématique. Pour avoir la partie fractionnelle de a faire : −INTa

DEG, DMS

L'utilisation la plus courante de ces fonctions est, sur le PC-1500, dans le calcul d'un temps.

T1=TIME
. . . .
T2=TIME
DMS (DEGT2−DEGT1)
Vous donnera le temps écoulé entre T1 et T2 sous la forme : MMJJHH. MMSS

APPLICATION A L'ETUDE DES FONCTIONS MATHEMATIQUES

Passage en notation scientifique.
Dans certaines sciences, notamment physique ou chimie, on travaille avec des puissances de dix multiples de 3. Par exemple :

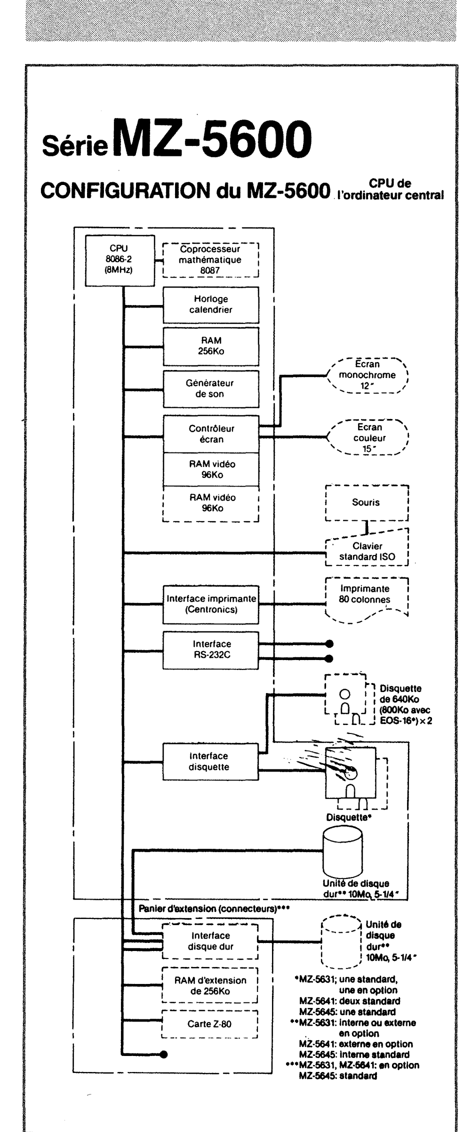
10^{-3}= milli 10^3= Kilo
10^{-6}= micro
10^{-9}= nano
10^{-12}= pico

Or aucun ordinateur de la gamme SHARP ne fait cette conversion d'origine, on va donc leur apprendre. Soit dans R la valeur à convertir :
PRINT R/10↑(INT(LOGR/3)*3);"E";INT(LOG R/3)*3
Mais cette formule a un défaut, par exemple, pour R=145, elle sortira 145E0 ce qui est juste, mais on préférera peut-être.
.145 E3 qui est plus couramment utilisé
S=INT (LOG R/3)*3
S=S+3*((INT LOG(R/10↑S)+1)/3=INT((INT LOG(R/10↑S)+1)/3))
PRINT R/10↑S;"E";S

Marc GIRONDOT

MZ 5600 LE MINI MULTI-POSTES MULTI-UTILISATEURS

Le MZ 5600 est une unité centrale présentant la même architecture qu'un mini-ordinateur. Il est orienté, en priorité, vers les applications multi-postes et multi-tâches.





1) LES MICROPROCESSEURS

L'élément principal est un **iAPX-86** fonctionnant à 8MHz. assisté par un contrôleur de BUS I.8089 et par 2 processeurs optionnels : I.8086-2 et I.8087-2 spécialisés dans les calculs.

• 4 microprocesseurs spécialisés prennent en charge la gestion de l'écran ; texte, graphisme haute résolution et couleurs.

• 1 processeur « **CANAL** » gère les transferts **RAM-RAM** et **RAM-DISQUE** par 4 cannaux totalement indépendants.

• 1 processeur gère indépendamment les unités de disquettes souples.

• Les entrées-sorties V24 sont également gérées, 2 par 2, par un processeur spécialisé.

• Même gestion spécialisée pour la sortie imprimante parallèle **CENTRONICS.**

2) LA MEMOIRE

• Les **256 Ko.** ou **512 Ko.** de mémoire de travail sont dotés d'une logique de protection de la mémoire qui évite le chevauchement des programmes en cas d'erreur sur l'un des programmes. Elle contrôle également le droit de lecture / écriture sur un espace mémoire donné. Ainsi, un programme en cours ne peut pas écrire ou même lire l'espace mémoire alloué à une autre application. **Cela élimine** toute possibilité de « **SCRATCH** » du système dans un environnement multi-utilisateurs.

• La **mémoire vidéo** standard a une capacité de **96 Ko.** extensible à **192 Ko.**

3) LES PERIPHERIQUES

• **Disquettes 5 Pouces** formattées à 800 Ko. ou 640 Ko. selon le système d'exploitation utilisé.

• **Disques durs** intégrés de 10 ou 20 Mo.

• Disques durs externes de 40 ou 64 Mo. avec possibilité de sauvegarde par « **STREAMER** » en quelques minutes.

• **Disques** souples de **8 pouces** d'une capacité unitaire de 1,2 Mo.

• Gamme d'**écrans** monochromes ou couleurs graphiques.

• **Terminaux** écran-clavier orientés caractères.

• Processeurs sur cartes permettant l'émulation de systèmes 8 bits.

• Périphériques d'entrée : Clavier ou **souris.**

LE MZ-5600 AU SICOB



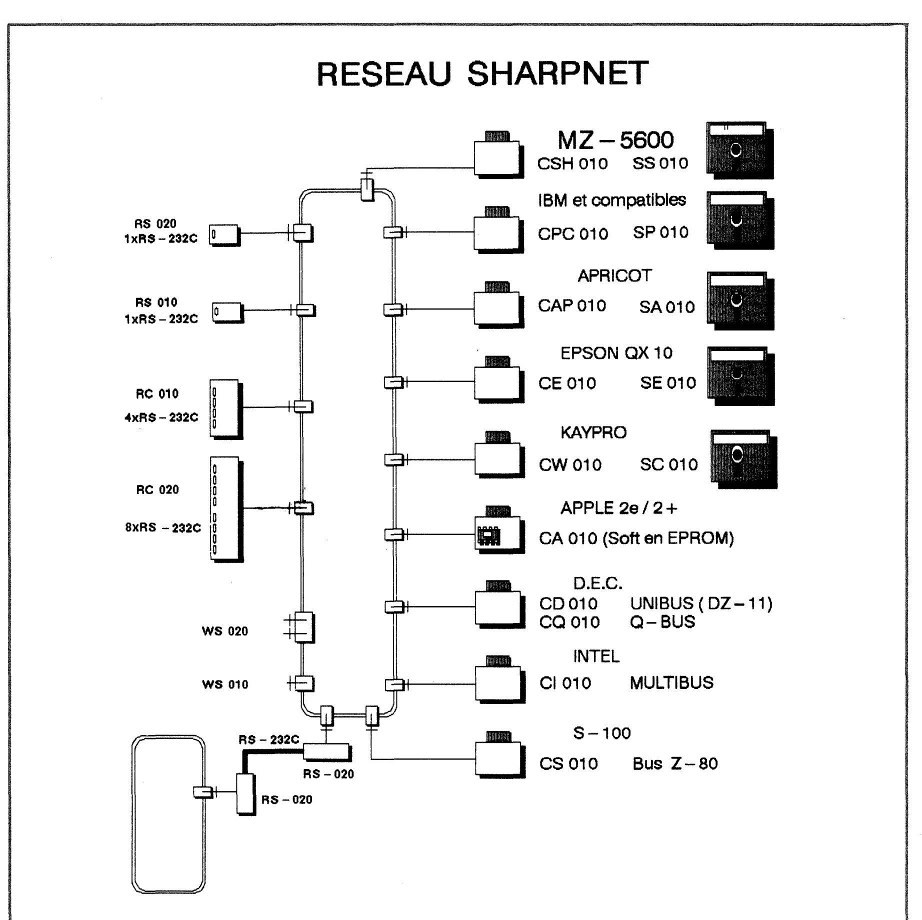
Terminal à écran

La politique de Sharp envers le MZ-5600 fut présentée au **SICOB 1985.**
Elle s'oriente selon trois grands axes qui sont :
- **La pluralité des « Operating Systèmes »**
- **Le développement des communications**
- **Le poste de travail**

7 ateliers disticts ont composé le stand Sharp Système
LE SERVEUR MINITEL ou l'on pouvait utiliser la méssagerie électronique, un programme de gestion de rendez-vous et plusieurs autres applications.
Deux minitels connectés par lignes téléphoniques à un MZ56/60 (MZ-5600 avec disque dur de 67Mo.) parmettait l'intégration du site serveur. Localement un écran permettait la mise à jour des différents renseignements.
LE MULTIPOSTE composé de 6 écrans/clavier (terminaux intelligents) connectés sur un site central (disque dur 67Mo. et sauvegarde par « streamer » de 60Mo.) ; d'une imprimante qualité listing ou courrier, et d'une liaison V24 (RS.422) connectée à une seconde unité centrale en réseau.
Le MZ-5600 a ainsi géré 8 tâches distinctes dont une : le superviseur multi-tâches qui joue le rôle de chef d'orchestre.
Le PC-7000 ordinateur personnel compacte et portable, dont la comptabilité lui permet de reprendre à son compte toute la bibliothèque des logiciels du PC-IBM. (voir notre rubrique nouveautés).

LE MZ-5600 FONCTIONNANT EN RESEAU

SHARPNET est un réseau en anneau compatible « MULTILINK » qui permet d'homogéiniser un environnement micro-informatique qui a grandi de façon hétérogène.
SHARPNET permet de connecter ensemble les systèmes les plus divers comme :
APPEL, IBM, COMMODORE, EPSON, KEYPRO, VAX, PDP 11, ainsi que toutes les machines possèdant les standars UNIBUS, Q.BUS, MULTIBUS et BUS S.100.

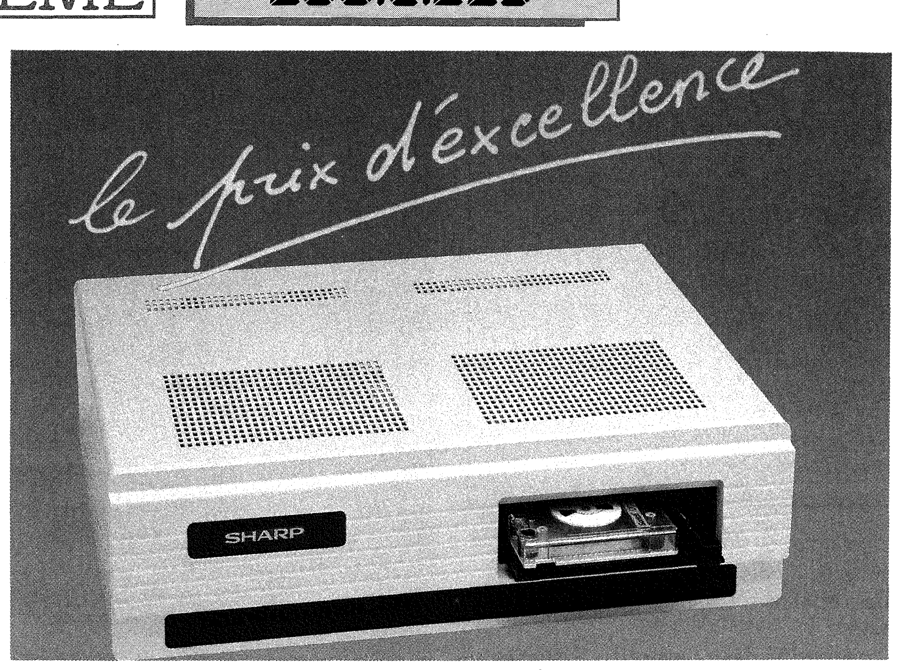


XNET un réseau intégré au système d'exploitation M.O.S.
Reliant par un canal rapide deux unités centrales, il gère la liaison en mode MULTI-TACHES ; deux ou plusieurs programmes peuvent ainsi s'exécuter simultanément et accèder aux ressources, disquettes, disques durs, imprimantes etc... de l'autre unité centrale.
Le poste 6 a rassemblé toutes **LES APPLICATIONS du type TABLEUR, BASE DE DONNEES, TRAITEMENT DE TEXTE** en présentant les « Best-sellers » sous système d'exploitation MS.DOS.2.11.
LE POSTE DE TRAVAIL BUREAUTIQUE où fut présenté GEM (GRAPHIC ENVIRONNEMENT MANAGER) et ses logiciels intégrés. Répondant au besoin de simplification de l'interface HOMME/MACHINE, SHARP répond également au futurs besoins des consommateurs.

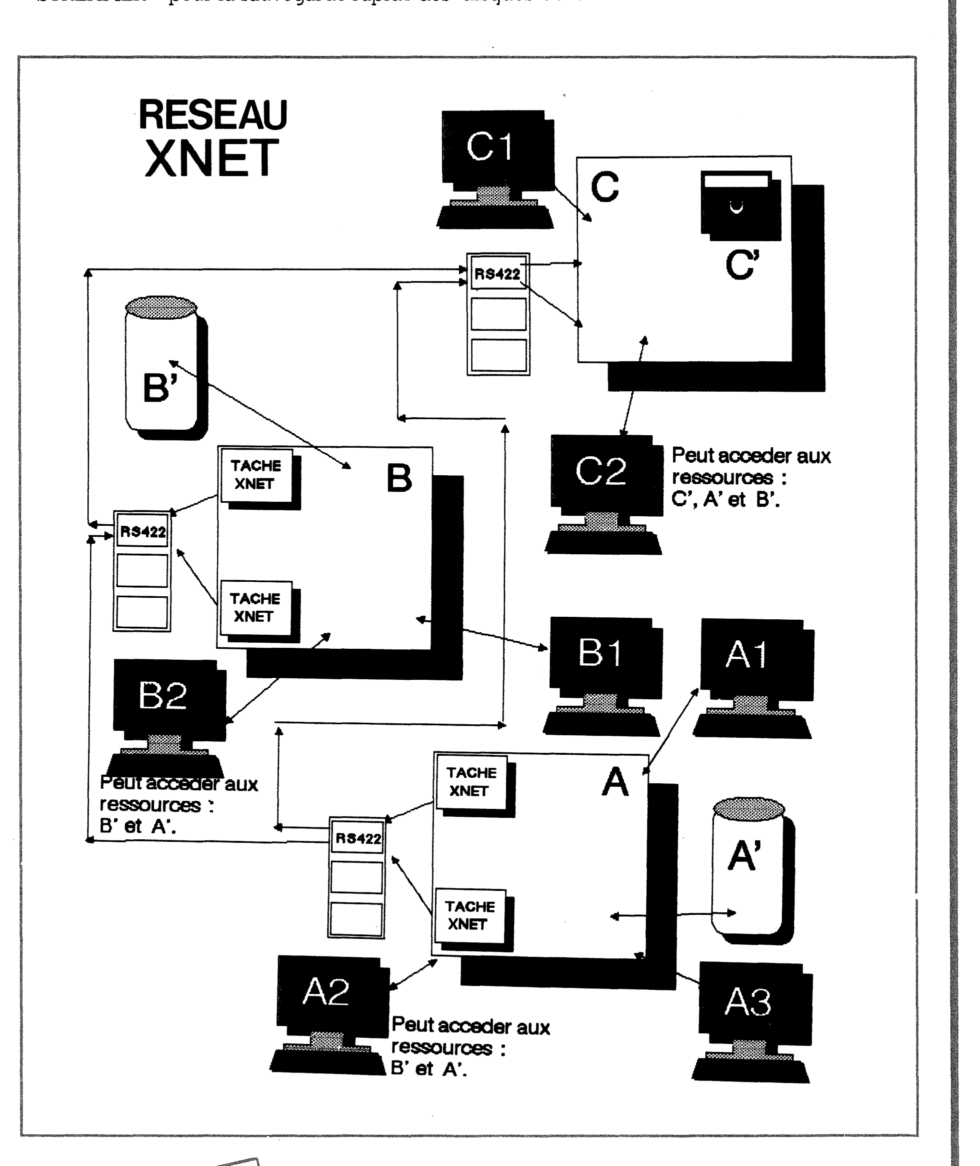
• Graphisme haute résolution (640 x 400 points).
• Graphisme couleur (8 couleurs par point).
• La souris et les icônes.

La présentation de GEM et ses logiciels intégrés : WRITE, DRAW, PAINT, fut une première mondiale très remarquée, même par nos visiteurs japonais.
Le second point d'attrait de ce poste fut un autre logiciel intégré :
LOGISTIX : **tableur graphique/Base de données/Gestion de projet.**
Compatible avec les fichiers des logiciels LOTUS 1,2,3, D BASE, SUPER CALC, VISI CALC, WARDSTAR, MAIL MERGE, etc, il permet de travailler sur 1024 x 2048 cases et de réprésenter des graphiques sous une résoution de 640 x 400 points.
Ces graphiques peuvent être reproduits sur l'imprimante à jet d'encre 10.700 ou sur le traceur de courbes CE.516P.
Le dernier poste de cette présentation a été dédié à la CAO et la DAO représentées respectivement par SHARPCAD et UNIPAINT.
SHARPCAD est un logiciel couleurs de conception assistée par ordinateur présentant des qualités similaires au Best-seller « AUTO-CAD » avec certaines caractéristiques plus évoluées.
UNIPAINT, présenté dans une version de test a les fonctionnalités des logiciels de la catégorie « Paint ». Il utilise pleinement les capacités des quatre micro-processeurs du MZ dédiés à la gestion graphique 640 x 400 points en 8 couleurs.
Deux points importants resteront gravés dans la mémoire des visiteurs de notre stand SYSTEME du SICOB : l'épanouissement des technologies et des applications professionnelles SHARP,... et la chaleur tropicale qui y régnait !

E. BERNARD



« STREAMER » pour la sauvegarde rapide des disques durs.



SYSTEME D'EXPLOITATION SHARP-5600

SYSTEME MOS

MERCURE OPERATING SYSTEM
performant, le Système **MOS** allié à la simplicité d'utilisation des micros, donne une étonnante puissance digne des plus gros systèmes. Le Système d'exploitation est de développement professionnel, conçu dans un esprit d'indépendance vis-à-vis du matériel (processeurs 8, 16, 32 bits...), modulaire et évolutif (mono et multi tâches, mono et multi-postes, mono et multi-processeurs : possibilité d'interconnexion de plusieurs ordinateurs de marques identiques ou différentes), puissant (fichier de plusieurs milliards de caractères, adaptables à tous types de périphériques), complet (partage des fichiers et des enregistrements, confidentialité des programmes et des fichiers, gestion de fichier séquentiel indexé multiclés (jusqu'à 20), tri de fichiers, langages interprétés et compilés : **MICROBOL** et **M**, éditeur pleine page, etc., (une trentaine d'utilitaires en version de base), simple d'emploi (conversationnel et en français), documenté (classeur de 350 pages)...

XMMPP. Mise en page paramétrable.
Conçu pour des non informaticiens, ce module vous permet de composer vous-même l'édition et la mise en page de vos états imprimés. Indépendant du programme, il offre à l'utilisateur la plus grande souplesse d'utilisation, cette procédure est couramment appelée « paramétrage ».

XGEFI. Générateur de fichiers.
Créé pour les utilisateurs non informaticiens, XGEFI est un remarquable générateur de programmes de gestion de fichiers (création, interrogation multi-critères, mise à jour, sélection, tri, étiquettes, listes, mailing, etc.), il vous permet de créer des fichiers (prospects, clients, articles, représentants, etc.).

XLOCK. Verouillage de disquettes.
Il interdit la duplication frauduleuse de programmes en « marquant » la disquette d'un fichier induplicable.

AIDES A LA DECISION BUREAUTIQUE

XPLAN
Un tableur à la hauteur de ses ambitions. Entièrement conversationnel la mise en œuvre d'XPLAN est un jeu d'enfant. Il permet de gérer des tableaux de 999 lignes × 999 colonnes. XPLAN effectue des calculs sur des valeurs numériques de 20 chiffres significatifs.
Il permet d'employer des fonctions : racine carrée, exponentielle, logarithme, sinus, cosinus, arc sinus, etc.
XPLAN permet la gestion de fenêtres indépendantes. Il peut récupérer des informations dans les fichiers de la comptabilité par exemple et éditer des résultats à partir de ces données.

CTEXTE
Traitement de texte simple d'emploi répondant aux besoins les plus courants.
L'éditeur CTEXTE comporte deux modes :
- le mode « saisie »
- le mode « commande »,

il autorise la frappe au « kilomètre ». Avec des fonctions de centrage, de justification à droite, de double frappe, soulignement, tabulation, saut de page, etc.
Il permet d'éditer des textes pour mailing à partir d'un fichier de référence.
Sa simplicité d'utilisation est remarquable.

LATIN
Le LATIN (langage de Traitement des Informations) est le langage naturel de MERCURE.
Il a été créé : pour les non-informaticiens qui peuvent sélectionner des informations de leurs fichiers, les afficher, les imprimer, les extraire dans un autre fichier (pour le traitement de texte par exemple) ou les mettre au format du tableur.
Pour les informaticiens, réduire les temps d'écriture de programmes de sélection, liste, interrogation (réduction des délais et des coûts).
Un programme LATIN est constitué de CHAPITRES. Chaque chapitre est constitué d'instructions en français.
Un outil puissant et simple d'utilisation.

MINITEL ET COMMUNICATIONS

XIMA. Composition d'images sur MINITEL.
Il permet de composer, à partir d'un clavier-écran normal, une image d'écran MINITEL et de la stocker dans un fichier.

XMES. Messagerie
Les messages individuels et collectifs sont stockés par destinataire jusqu'à ce que ce dernier en ait eu connaissance.

XDOC. Banque de données à Recherche documentaire
Ce logiciel vise de façon générale à faciliter les échanges d'informations dans un groupe de personnes concernées par un dommaine commun.
Il gère une centaine de « banques ». Une banque contient des textes ou des données avec des mots clés. Orienté MINITEL, peut s'utiliser avec des écrans normaux.
Son domaine d'application déborde de la recherche documentaire pour traiter par exemple, l'aide au diagnostic, la gestion de textes tels des notes de service, archivage de secrétariat, etc.

XCOM. Télétraitement.
A travers un micro-ordinateur local, un programme peut être exécuté sur un autre ordinateur (éventuellement à distance par MODEM) en utilisant l'écran-clavier local, l'imprimante locale et les fichiers de deux ordinateurs.

Indispensable :
- pour le transfert de fichiers et leurs mises à jour dans des sociétés à implantations multiples.
- pour le transfert des fichiers sur des matériels différents (réseau hétérogène).
- pour la télé-assistance entre l'installateur de logiciels et ses clients.

COMPTABILITE GENERALE

4 000 INSTALLATIONS DONT PLUSIEURS CENTAINES DE CABINETS COMPTABLES.
La comptabilité MERCURE est multi-dossiers : Multi-sociétés tenues simultanément, la durée de l'exercice n'est pas limitée à 12 mois. Comptabilités auxiliaires clients et fournisseurs, fonctions multi-utilisateurs (saisie des écritures, de plan comptable, consultations, édition grand livre et balance)... Les comptes sont créés ou modifiés en cours d'année. Numérotation automatique, solde progressif de chaque pièce, journaux : bilan d'ouverture, achats, ventes, banques et caisse, opérations diverses, régularisation et inventaire, centralisation de comptes à la demande.
Lettrage manuel ou automatique. Edition des comptes, trois options : début de l'exercice, de la période, des écritures non lettrées.
Balance détaillée et balance récapitulative, bilan, comptes de résultats avec comparaison multi-exercices, clôture, réouverture, nouvel exercice. Branchement pour comptabilité analytique. Edition claire, facilement lisible sur listings pré-imprimés.
D'une grande simplicité grâce à sa saisie entièrement conversationnelle, le logiciel de comptabilité MERCURE ne nécessite pas de connaissance informatique de la part de l'utilisateur.

GESTION COMMERCIALE

INSTALLE A PLUS DE 2 000 EXEMPLAIRES, CE LOGICIEL EXTREMEMENT COMPLET, REPOND A LA QUASI-TOTALITE DES BESOINS DES P.M.E.-P.M.I., ETC.
GESTION DES COMMANDES

Faire appel aux fichiers clients et articles. Possibilité de consultation, modification, suppression, édition d'une commande saisie.
Validation totale ou partielle avec gestion des reliquats déclenchant la facturation automatique.

FACTURATION DIRECTE
Indépendante du carnet de commande, elle fait également appel aux fichiers clients et stocks avec mise a jour automatique.
Possibilité d'éditer des traites. Journal des ventes automatique avec intégration dans les comptes de tiers et produits.

STATISTIQUES DIVERSES
Possibilité d'extraire tous types de statistiques (représentants, produits, etc.).

STOCK
Gestion du stock physique. Edition d'état détaillé des mouvements. Consultation, inventaire, valorisation, gestion des entrées et sorties.
Calcul de la marge brute par article et famille.
Possibilité de connexion avec la comptabilité générale.

PAIE

PERFORMANTE.
3 000 INSTALLATIONS DONT UNE CENTAINE EN CABINETS COMPTABLES
La paie MERCURE est multi-entreprises, les règles de calcul, de comptabilisation et d'édition de journal de paie sont entièrement paramétrables avec possibilité de modifier les règles de calcul de chaque rubrique ou de créer de nouvelles rubriques.
Tous types de paie peuvent être pris en compte : abattements (ouvrier, VRP, apprenti), bâtiment (intempérie), repos compensateur, congés payés, caisse de retraite sans limitation de nombre.
Les fonctions réalisées sont : Gestion de fichiers, entreprise et salariés, calcul et édition du bulletin de paie, journaux de paie et des opérations diverses, annulation de bulletin de paie, calcul fictif de paie, liste par mode de règlement, DAS, AAS,etc..
Possibilité de connexion avec la comptabilité générale.

LOGICIEL F.N.E.E.

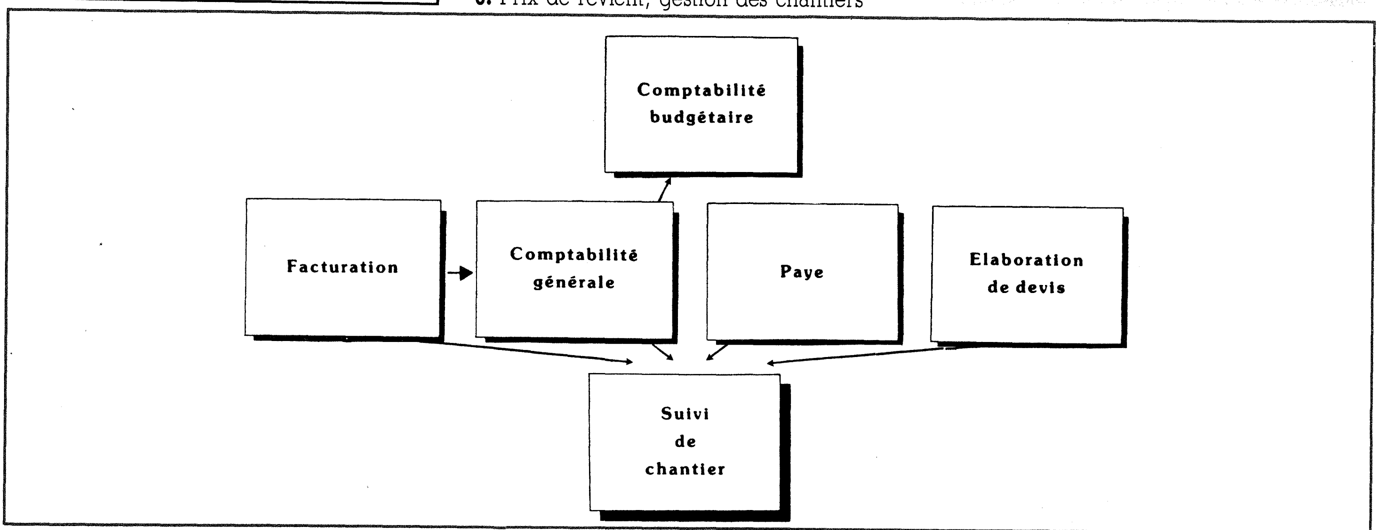
DES LOGICIEL DE GESTION ET DEVIS POUR ELECTRICIENS.

La Fédération Nationale de l'équipement électrique a conçu des logiciels pour assurer toutes les fonctions informatiques d'une entreprise d'équipement électrique.

Ces logiciels sont mis gratuitement à la disposition des entreprises affiliées **FNEE**, qui ne payent que des frais d'installation, et sont en permanence tenus à jour.

LEURS FONCTIONS :
1. Paye
2. Comptabilité
3. Prix de revient, gestion des chantiers
4. Facturation
5. Devis

Ces logiciels se caractérisent par leur intercommunication et diminuent sensiblement le volume des saisies ainsi qu'évitent les erreurs dues aux reports manuels d'écriture.



Programme 1. Paye
- Gestion des salaires
- Paye « ouvriers » par simple saisie de feuilles d'heure
- Imputation automatique

Programme 2. Comptabilité-gestion
Ce programme assure la fourniture des documents nécessaires à la comptabilité générale du plan comptable 1982 et est caractérisé par la rapidité de saisie des fichiers clients et fournisseurs.
Il assure aussi la possibilité de comparer mois par mois ses prévisions avec la réalité des comptes comptables.
- Analytique :
Permet lors de la saisie des factures fournisseur de donner un numéro de chantier permettant ainsi l'affectation de toutes factures directement sur les comptes chantier.

Programme 3. Prix de revient-gestion des chantiers
Logiciel rassemblant toutes les données analytiques pour sortir un compte chantier comprenant :
- nombre d'heures et leur coût
- débours matière
- total des débours
- frais indirects imputés au prorata des heures
- facturation
- écarts

Programme 4. Facturation
- édition de l'entête client
- enregistrement montant HT
- calcul TVA et du TCC
- comptabilité

Programme 5. Devis
- une codification article
- désignation
- mention du fournisseur
- mention du produit
- mention du temps de pose
- 3 volumes de prix.

EXPERTISE AUTOMOBILE
FONCTIONALITES
• Enregistrements des dossiers, interrogation • Editiondes convocations • Liste des sessions • Saisie des rapports et impression • Listes des pièces détachées • Notes d'information • Chiffrage et impression des honoraires • Editions des relevés et saisie des règlements • Statistiques dossiers non traités • Entrées et sorties périodiques • Archivage • Modification des références • Fichier des pièces détachées

FACTURATION GENERALE
FONCTIONALITES
Logiciel de la facturation en liaison aux fichiers clients et articles.
Il assure la facturation en temps différé, le journal des ventes, l'inventaire en quantité et en valeur.
Liaison avec la comptabilité lors du journal des ventes

OPTION :
Gestion des commandes
Les commandes sont enregistrées en mode conversationnel. Le carnet de commande interrogé par client, par article. Les commandes sont facturées automatiquement.

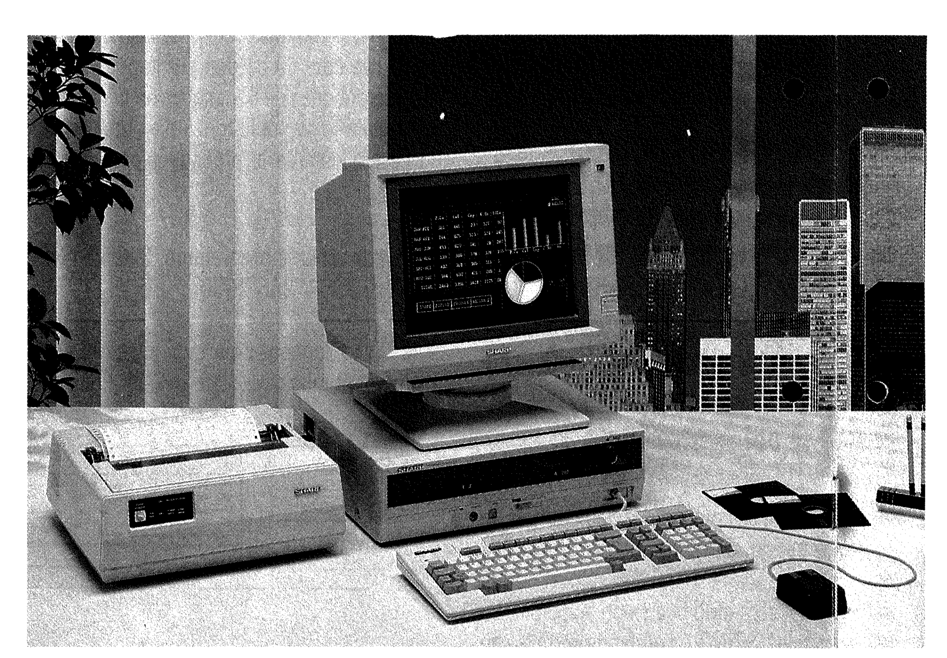
GESTION DE PRODUCTION
FONCTIONALITES
Description :
Gestion des fichiers :
• Matière première avec valorisation et fournisseur • Pièces détachées avec matière usinage et sous traitance • Fournitures extérieures avec prix achat et prix vente • Sous ensemble, ensemble, client et entreprise
Suivi de la fabrication suivant :
• Fabrication pièces détachées • Fabrication des sous ensembles • Saisie des commandes d'ensembles
Pour chacune de ces phases :
• Lancement, entrée en stock et liste • Facturation et journal des ventes

GESTION DE CHANTIERS
FONCTIONALITES
• Création du fichier Chantiers • Saisie des mouvements avec édition sur option de la commande fournisseur • Gestion du fichier tarif • Gestion du fichier fournisseur • Saisie et édition des devis • Saisie en version société • Saisie en version client

S.A.V.
FONCTIONALITES
• Saisie des appels (fichier client) • Edition des appels • Saisie des causes de panne • Consultation de l'historique par appareil et édition • Liaison avec la facturation gestion de stock • Liaison avec la comptabilité

CABINETS DENTAIRES
FONCTIONALITES
• Gestion du fichier client avec passe médicale, contre indications, shéma dentaire • Saisie des soins, des travaux de prothèse, des travaux ODF, des acomptes de règlements • Impression feuille de SS, fiche de soins, de prothèse, ODF, et devis • Gestion automatique du tiers payant • Gestion des impayés et relancés • Gestion du secrétariat • Fonction agenda et fonction courrier • Gestion comptable • Gestion administrative

CABINETS VETERINAIRES
FONCTIONALITES
• Suivi fiche client • Edition historique par animal • Lettre de rappel date de vaccin • Editions et relance des impayés • Suivi des règlements • Gestion du fichier client • Edition journal des ventes • Traitement comptabilité



LES NOUVEAUTES

EPF 1500

UN PROGRAMMATEUR D'EPROM INTELLIGENT POUR PC 1500

L'EPF 1500 et un périphérique intelligent connectable sur le PC 1500. Ce produit est un programmateur d'EPROM utilisable comme une mémoire de masse au même titre qu'une disquette. Le temps de chargement est cependant inférieur à celui d'une disquette. Cette fonction permet à l'utilisateur de constituer une bibliothèque de programmes BASIC ou LM, ou de fichiers sur silicium ; support de faible coût et résistant aux environnements sévères. Dans un autre contexte, l'EPF 1500 est capable de générer des textes hexadécimaux grace à son éditeur et de les transcrire sur une EPROM servant par exemple de générateur de caractères, de moniteur, de générateur de systèmes automates, etc. L'EPF 1500 connecte l'EPROM directement sur le bus du microprocesseur, les mémoires utilisables sont : 27128, 2764, 2732A, 2732, 2716. La connection est possible avec un PC 1500 OU 1500A muni ou non de l'interface CE 150. Un jeu d'instructions BASIC spécifiques permet une manipulation très simple de ce programmateur.

L'EPF 1500 est disponible, en exculusivité au Club des SHARPENTIERS au prix de 2 200 F. TTC port compris (règlement à l'ordre de société S.B.M.).

2 200F

PC BLUES

Le programme PC BLUES est un programme de création musicale. Il est étonnant de l'entendre jouer une partition, le résultat est vraiment superbe. D'origine ce programme ne pouvait sortir des sons que sur le buzzer, ce qui donnait de très bons résultats, mais le niveau sonore était un peu faible, c'est pourquoi il a été adapté sur le haut-parleur d'un magnéto, et le résultat est fantastique !

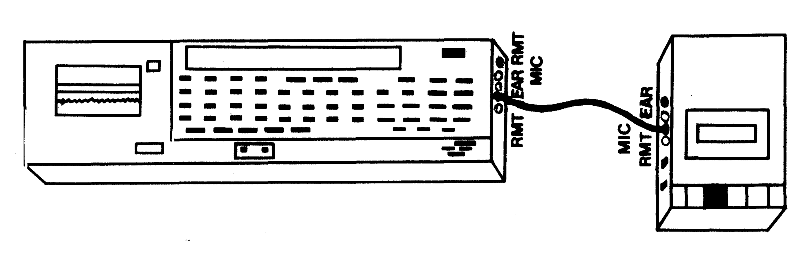
Le programme se compose de deux parties, la première permet de rentrer une partition. Elle est identique dans tous les cas. C'est la seconde qui sera modifiée selon le besoin. La version sur le LISTING 1 permet de sortir le son sur le buzzer du PC-1500. Si vous modifiez le LISTING 1 en tapant le 2, vous allez sortir le son sur un haut-parleur externe, on est obligé de réécrire le programme qui gère le BEEP pour faire sortir le son sur la sortie EAR. Il faudra alors brancher le PC-1500 sur le CE-150, brancher un des cordons sur la prise EAR du CE-150 et l'autre côté du cordon sur la prise MIC du magnétophone que l'on mettra en mode PLAY, en réglant le son et la tonalité en médium. On l'utilisera comme le précédent pour sortir le son.

Avant de taper le programme, faites un NEW PEEK & 7864*256-2000.

Tapez ces données pour avoir un exemple de différentes musiques (LISTING 3). Pour avoir accès à cette nouvelle fonction à partir du basic, tapez, après avoir fait NEW PEEK & 7863*256+&1C5 en mode PRO, le programme du LISTING 4, et faites RUN, il se chargera automatiquement. Pour avoir accès à la nouvelle fonction : DEF P.

Il y a alors deux possibilités suivant que vous avez une ancienne ou une nouvelle ROM :
• clavier sonorisé – nouvelle ROM (en BEEPON et sur imprimante) DEF P affichera PLAY.
• clavier non sonorisé – ancienne ROM tapez sur la touche ! qui est affectée automatiquement puis faites DEFP qui affichera PLAY.

La syntaxe de cette nouvelle instruction est : PLAY, n, f, d les paramètres sont à rapprocher de BEEP n,f,d,
où f est la fréquence 0-255



d est la durée 0-65535
et n, est le nombre de BEEP 0-65535
Vous remarquerez que la fonction LIST a disparu puisque PLAY a le même code qu'elle. Pour y avoir quand même accès, faites : OPN ENTER
LIST n ou L.n comme d'habitude.

Voici enfin un exemple de musique gérée par le basic pour admirer la différence :
DEF A — sortie sur le buzzer
DEF B — sortie sur le magnéto
LISTING 5
Pour le commander en langage machine :

BASIC	buzzer	haut-parleur externe
BEEP 1	SPJ&E669	SPJ[PEEK&7863-I)*256+1]
BEEP 1,UI	SPJ&E66B	SPJ[PEEK&7863-I)*256+3]
BEEP 1,UI,X	SPJ&E66F	SPJ[PEEK&7863-I)*256+7]

Si vous avez un PC-1500 ancienne ROM, un ON-OFF affichera :
NEW 0 : CHECK
n'en tenez pas compte, tapez :
CL
puis ! pour initialiser

MODE D'EMPLOI DE PC BLUES

NEW PEEK&7864*256-2000
Charger ou taper le programme
CLEAR ENTER
DEF A
* affichage note ?

touche :
SHIFT SI
! DO
" RE
// MI
$ FA
% SOL
& LA
* 4 3 2 1.51.5
correspond à la durée :
SHIFT ! " // $ 9 8 DEF OFF
* BASSE --- MEDIUM --- AIGU
! " // $ % & OFF
DEF S
Sauvegarde des notes sur cassette
DEFL
Rechargement des notes
RUN
Sortie des notes sur le buzzer

ASTUCE

Pour rendre compatible GALACTICA de APSOFT avec la sortie de son sur magnétophone externe, faites après chargement :
POKE PEEK&7863*256+&3F6,&BO,&OE,1
POKE
PEEK&7863*256+&3FD,&BO,&OE,&FE
et faites le branchement indiqué dans l'article PC BLUES M. La voix sortira sur le haut-parleur.

Pour avoir un meilleur rendu sonore, il semble préférable de faire :
POKE PEEK&7863*256+&3F9,&38,&38 et de mettre la tonalité sur le grave et le son au maximum.

LISTING 1

```
1 "P"REM              LOUTTER Michel
2 POKE 29008,&68,PEEK &7863,&6A,&C5,&25,&89,&01,&9A
3 POKE 29016,&0A,&64,&25,&89,&01,&9A,&08,&64,&25,&38
4 POKE 29026,&89,&01,&9A,&FD,&A8,&2A,&BE,&E6,&6F
5 POKE 29035,&62,&BE,&E6,&6F,&62,&BE,&E6,&6F
6 POKE 29043,&60,&BE,&E6,&6F,&62,&BE,&E6,&6F
7 POKE 29051,&FD,&2A,&BE,&E4,&51,&8B,&01,&9A,&64,&9E,50
8 WAIT 0:PRINT "PC. BLUES";
10 CALL 29008:END
50 "A"WAIT 0:CLS :IF X<PEEK &7863*256+&C5LET X=PEEK &7863*256+&C5
51 IF P=0LET P=1
52 PRINT A$;L;" ";M;" ";X;
53 GPRINT "0000FFFF3FFFFF5FFFFF6FFFFF77FFFF7BFFFF7DFFFF7EFFFF";
54 PRINT " note?";:ON ERROR GOTO 200
55 BEEP 1,4,5:A$=INKEY$ :IF A$=""GOTO 55
56 GOSUB (ASC A$)+56:PRINT A$;:BEEP 1,E,90:GOTO 80
57 E=81:T=99:A$="SI":RETURN
73 E=154:T=52:A$="DO":RETURN
74 E=137:T=59:A$="RE":RETURN
75 E=122:T=66:A$="MI":RETURN
76 E=115:T=70:A$="FA":RETURN
77 E=102:T=78:A$="SOL":RETURN
78 E=91:T=88:A$="LA":RETURN
80 IF P=.95CLS :PRINT "# 4 # 3 # 2  1.5  1 # .5";
81 IF P=1CLS :PRINT "  4   3   2  1.5  1   .5";
84 ON ERROR GOTO 201
85 BEEP 1,1,1:C$=INKEY$ :IF C$=""GOTO 85
86 IF ASC C$=27LET P=.95:GOTO 80
87 IF ASC C$=15LET P=1:GOTO 80
90 GOSUB (ASC C$)+90:GOTO 130
91 L=.25:RETURN
107 L=4:RETURN
108 L=3:RETURN
109 L=2:RETURN
110 L=1.5:RETURN
111 L=1:RETURN
112 L=.5:RETURN
130 BEEP 1,E,50:CLS :PRINT "BASSE------MEDIUM----AIGU";
131 BEEP 1,1,1:E$=INKEY$ :IF E$=""GOTO 131
132 IF (E$=CHR$ 20)OR (E$=CHR$ 19)LET M=1:GOTO 140
133 IF E$=CHR$ 15LET M=.25:GOTO 140
134 IF E$=CHR$ 22LET M=.5:GOTO 140
135 IF E$=CHR$ 17LET M=1.5:GOTO 140
136 CLS :PRINT "ERREUR!";:BEEP 1,60,300:CLS :GOTO 130
140 BEEP 1,E,10
141 E=INT (P*E*M):H=INT (T*L*(1/M)):H=H*.5:O=H/255:W=INT O:Z=INT ((O-W)*255)
142 REM WAIT 40:PRINT E;"  ";W+1;"  ";Z:WAIT 0
150 POKE X,Z,W+1,E,0,0,0:X=X+3:IF X+5>=PEEK &7865*256+PEEK &7866BEEP 5:WAIT :P
RINT "FULL UP":END
152 GOTO "A"
180 "S"BEEP 1,200,200:CLS :WAIT 0:PRINT "CSAVING  PC BLUES";
181 PRINT #"PC BLUES";X
182 BEEP 1,200,800:CSAVE M"PC BLUES";PEEK &7863*256+&C5,X+1
200 CLS :PRINT "ERREUR !";:BEEP 1,70,400:GOTO 52
201 CLS :PRINT "ERREUR !";:BEEP 1,60,300:GOTO 80
500 "L"WAIT 0:BEEP 1:PRINT "LOAD PC BLUES"
501 INPUT #;X:CLOAD MPEEK &7863*256+&C5:END
```

LISTING 2

```
1 "P"REM              LOUTTER Michel
2 WAIT 0:PRINT "PC. BLUES";:RESTORE 30:A=(PEEK &78BE-128)*256+PEEK &78BF+1
3 POKE &7750,&68,PEEK &7863,&6A,&C5,&25,&89,&01,&9A
4 POKE &7758,&0A,&64,&25,&89,&01,&9A,&08,&64,&25,&38
5 POKE &7762,&89,&01,&9A,&FD,&A8,&2A,&BE,A/256,AAND &FF
6 POKE &776B,&62,&BE,A/256,AAND &FF,&62,&BE,A/256,AAND &FF
7 POKE &7773,&60,&BE,A/256,AAND &FF,&62,&BE,A/256,AAND &FF
8 POKE &777B,&FD,&2A,&BE,&E4,&51,&8B,&01,&9A,&64,&9E,50
10 POKE A,&FD,&98,&FD,&88,&FD,&A8,&58,&B0,&5A,&0E,&24,&28,&B5,&01,&FD,&1E,&A4

11 POKE A+17,&2A,&88,&02,&FD,&ED,&F0,&0B,&03,&89,&14,&46,&84,&8B,&10,&B5,&00
12 POKE A+33,&FD,&1E,&A4,&2A,&88,&02,&6A,&01,&88,&02,&49,&FF,&9E,&23,&FD,&2A,
&BA
13 POKE A+50,&E5,&AF
20 CALL &7750:END
30 "BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB"
```

LISTING 3

```
A=PEEK &7863*256       ENTER
POKE A+197,16,1,137,30,1,68,23,1,91,18,1,115,23,1,91,18     ENTER
POKE A+213,1,115,16,1,137,23,1,91,18,1,115,16,1,137,18,1     ENTER
POKE A+229,115,23,1,91,30,1,68,16,1,137,30,1,68,23,1,91     ENTER
POKE A+245,18,1,115,23,1,91,18,1,115,16,1,137,23,1,91,18     ENTER
POKE A+261,1,115,16,1,137,18,1,115,23,1,91,30,1,68,16,1     ENTER
POKE A+277,137,30,1,68,23,1,91,18,1,115,23,1,91,18,1,115     ENTER
POKE A+293,16,1,137,23,1,91,18,1,115,16,1,137,18,1,115,23     ENTER
POKE A+309,1,91,30,1,68,16,1,137,30,1,68,23,1,91,18,1     ENTER
POKE A+325,115,23,1,91,18,1,115,16,1,137,30,1,68,27,1,77     ENTER
POKE A+341,30,1,68,23,1,91,18,1,115,23,1,91,30,1,137,30     ENTER
POKE A+357,1,68,23,1,91,30,1,68,23,1,91,18,1,115,23,1     ENTER
POKE A+373,91,30,1,137,30,1,68,34,1,61,36,1,57,34,1,61     ENTER
POKE A+389,36,1,57,30,1,68,34,1,61,30,1,68,34,1,61,27     ENTER
POKE A+405,1,77,30,1,68,27,1,77,30,1,68,23,1,91,30,1     ENTER
POKE A+421,68,27,1,77,27,1,77,30,1,68,23,1,91,18,1,115     ENTER
POKE A+437,23,1,91,30,1,137,30,1,68,23,1,91,30,1,68,23     ENTER
POKE A+453,1,91,18,1,115,23,1,91,30,1,137,30,1,68,34,1     ENTER
POKE A+469,61,36,1,57,34,1,61,36,1,57,30,1,68,34,1,61     ENTER
POKE A+485,30,1,68,34,1,61,27,1,77,30,1,68,27,1,77,30     ENTER
POKE A+501,1,68,34,1,61,36,1,57,40,1,102,59,1,102,27,1     ENTER
POKE A+517,77,53,1,77,50,1,81,36,1,81,36,1,115,54,1,115     ENTER
POKE A+533,23,1,91,45,1,91,15,1,102,34,1,122,45,1,91,40     ENTER
POKE A+549,1,102,40,1,102,7,1,115,36,1,115,34,1,122,30,1     ENTER
POKE A+565,137,45,1,91,15,1,102,40,1,102,59,1,102,27,1,77     ENTER
POKE A+581,53,1,77,50,1,81,36,1,81,36,1,115,54,1,115,23     ENTER
POKE A+597,1,91,45,1,91,15,1,102,34,1,122,45,1,91,40,1     ENTER
POKE A+613,102,40,1,102,7,1,115,36,1,115,34,1,122,30,1,137     ENTER
POKE A+629,45,1,91,15,1,102,3,1,61,60,1,68,53,1,77,50     ENTER
POKE A+645,1,81,45,1,91,55,1,68,60,1,68,53,1,77,50,1     ENTER
POKE A+661,81,45,1,91,40,1,102,41,1,77,3,1,61,53,1,77     ENTER
POKE A+677,40,1,102,36,1,115,55,1,68,60,1,68,3,1,91,26     ENTER
POKE A+693,1,81,60,1,68,41,1,77,3,1,61,60,1,68,53,1     ENTER
POKE A+709,77,50,1,81,45,1,91,55,1,68,60,1,68,53,1,77     ENTER
POKE A+725,45,1,91,45,1,91,40,1,102,41,1,77,3,1,61,53     ENTER
POKE A+741,1,77,40,1,102,36,1,115,55,1,68,60,1,68,36,1     ENTER
POKE A+757,61,30,1,68,60,1,68,41,1,77,27,1,154,16,1,137     ENTER
POKE A+773,18,1,115,17,1,122,16,1,137,40,1,102,40,1,102,20     ENTER
POKE A+789,1,102,23,1,91,17,1,122,18,1,115,30,1,137,30,1     ENTER
POKE A+805,137,16,1,137,18,1,115,17,1,122,16,1,137,27,1,154     ENTER
POKE A+821,53,1,77,40,1,102,34,1,122,27,1,154,16,1,137,18     ENTER
POKE A+837,1,115,17,1,122,16,1,137,40,1,102,40,1,102,20,1     ENTER
POKE A+853,102,23,1,91,17,1,122,18,1,115,30,1,137,30,1,137     ENTER
```

```
POKE A+869,16,1,137,18,1,115,17,1,122,16,1,137,14,1,154,20   ENTER
POKE A+885,1,102,16,1,137,17,1,122,27,1,154,27,1,154,27,1   ENTER
POKE A+901,154,34,1,121,30,1,136,27,1,154,36,1,115,40,1,102   ENTER
POKE A+917,25,1,91,7,1,115,45,1,137,17,1,122,36,1,115,34   ENTER
POKE A+933,1,122,36,1,115,30,1,137,53,1,154,59,1,136,40,1   ENTER
POKE A+949,154,17,1,121,30,1,136,27,1,154,36,1,115,40,1,102   ENTER
POKE A+965,45,1,91,40,1,102,54,1,115,18,1,115,40,1,102,40   ENTER
POKE A+981,1,102,36,1,115,34,1,122,42,1,115,27,1,154,27,1   ENTER
POKE A+997,154,27,1,154,30,1,137,27,1,154,34,1,121,27,1,153   ENTER
POKE A+1013,27,1,154,27,1,154,36,1,115,40,1,102,45,1,91,45   ENTER
POKE A+1029,1,91,45,1,91,40,1,102,36,1,115,36,1,115,34,1   ENTER
POKE A+1045,122,30,1,137,30,1,137,30,1,137,54,1,115,34,1,122   ENTER
POKE A+1061,36,1,115,30,1,137,27,1,154,30,1,137,30,1,137,27   ENTER
POKE A+1077,1,154,34,1,121,30,1,136,27,1,154,27,1,154,27,1   ENTER
POKE A+1093,154,30,1,137,30,1,137,27,1,154,34,1,121,30,1,136   ENTER
POKE A+1109,27,1,154,36,1,115,40,1,102,45,1,91,45,1,91,45   ENTER
POKE A+1125,1,91,40,1,102,54,1,115,18,1,115,40,1,102,40,1   ENTER
POKE A+1141,102,36,1,115,34,1,122,42,1,115,14,1,154,16,1,137   ENTER
POKE A+1157,17,1,122,20,1,102,20,1,102,23,1,91,20,1,102,17   ENTER
POKE A+1173,1,122,14,1,154,16,1,137,17,1,122,17,1,122,16,1   ENTER
POKE A+1189,137,14,1,154,30,1,137,14,1,154,16,1,137,17,1,122   ENTER
POKE A+1205,20,1,102,20,1,102,23,1,91,20,1,102,17,1,122,14   ENTER
POKE A+1221,1,154,16,1,137,17,1,122,17,1,122,16,1,137,16,1   ENTER
POKE A+1237,137,27,1,154,36,1,115,36,1,115,23,1,91,45,1,91   ENTER
POKE A+1253,23,1,91,20,1,102,20,1,102,17,1,122,14,1,154,30   ENTER
POKE A+1269,1,137,14,1,154,16,1,137,17,1,122,20,1,102,20,1   ENTER
POKE A+1285,102,23,1,91,20,1,102,17,1,122,14,1,154,16,1,137   ENTER
POKE A+1301,17,1,122,17,1,122,16,1,137,16,1,137,27,1,154,17   ENTER
POKE A+1317,1,122,17,1,122,20,1,102,17,1,122,17,1,122,40,1   ENTER
POKE A+1333,102,17,1,122,20,1,102,27,1,77,26,1,81,23,1,91   ENTER
POKE A+1349,23,1,91,20,1,102,16,1,137,17,1,122,18,1,115,16   ENTER
POKE A+1365,1,137,16,1,137,17,1,122,36,1,115,16,1,137,18,1   ENTER
POKE A+1381,115,26,1,81,23,1,91,20,1,102,26,1,81,53,1,77   ENTER
POKE A+1397,14,1,154,14,1,154,27,1,77,23,1,91,18,1,115,40   ENTER
POKE A+1413,1,102,17,1,122,14,1,154,18,1,115,20,1,102,23,1   ENTER
POKE A+1429,91,40,1,102,14,1,154,14,1,154,23,1,91,18,1,115   ENTER
POKE A+1445,40,1,102,17,1,122,14,1,154,18,1,115,17,1,122,16   ENTER
POKE A+1461,1,137,27,1,154,20,1,102,40,1,102,20,1,102,17,1   ENTER
POKE A+1477,122,18,1,115,20,1,102,23,1,91,26,1,81,27,1,77   ENTER
POKE A+1493,23,1,91,26,1,81,20,1,102,36,1,115,36,1,115,30   ENTER
POKE A+1509,1,68,60,1,68,30,1,68,34,1,61,27,1,77,30,1   ENTER
POKE A+1525,68,26,1,81,27,1,77,26,1,81,23,1,91,20,1,102   ENTER
POKE A+1541,20,1,102,18,1,115,34,1,122,34,1,122,20,1,102,40   ENTER
POKE A+1557,1,102,20,1,102,17,1,122,18,1,115,20,1,102,23,1   ENTER
POKE A+1573,91,26,1,81,27,1,77,23,1,91,26,1,81,20,1,102   ENTER
POKE A+1589,36,1,115,36,1,115,30,1,68,26,1,68,30,1,68,34   ENTER
POKE A+1605,1,61,27,1,77,30,1,68,26,1,81,27,1,77,26,1   ENTER
POKE A+1621,81,23,1,91,20,1,102,27,1,115,34,1,122,50,1,122   ENTER
POKE A+1637,27,1,77,27,1,77,27,1,77,27,1,77,26,1,81,30   ENTER
POKE A+1653,1,68,30,1,68,27,1,77,26,1,81,23,1,91,23,1   ENTER
POKE A+1669,91,20,1,102,59,1,102,40,1,102,26,1,81,26,1,81   ENTER
POKE A+1685,26,1,81,26,1,81,0,0   ENTER
```

LISTING 4

```
10 DATA &55,&6A,&08,&48,&01,&4A,&A0,&FD,&98,&FD,&88,&FD,&A8,&58,&B0,&5A,2075
20 DATA &0E,&24,&28,&B5,&01,&FD,&1E,&A4,&2A,&88,&02,&FD,&ED,&F0,&0B,&03,1643
30 DATA &89,&14,&46,&84,&8B,&10,&B5,&00,&FD,&1E,&A4,&2A,&88,&02,&6A,&01,1429
40 DATA &88,&02,&49,&FF,&9E,&23,&FD,&2A,&BA,&E5,&AF,&B5,&00,&AE,&78,&63,2118
50 DATA &BE,&E2,&4A,&2A,&48,&01,&4A,&10,&BE,&28,&07,&6E,&0F,&89,&03,&BA,1383
60 DATA &E3,&3F,&8E,&8D,&B4,&50,&4C,&41,&59,&F0,&90,&28,&6C,&D0,&00,&00,1803
70 DATA &00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&DE,&71,&D2,&6F,656
80 DATA &80,&ED,&7A,&01,&80,&89,&69,&ED,&7A,&02,&F0,&8B,&63,&D0,&00,&5F,2000
90 DATA &FD,&A8,&C8,&2D,&B5,&FF,&FD,&2A,&FD,&98,&1A,&FD,&A8,&14,&8B,&18,2432
100 DATA &BE,&28,&01,&FD,&2A,&89,&42,&66,&24,&89,&03,&A4,&8B,&3B,&FD,&A8,1790
110 DATA &68,&00,&6A,&06,&CD,&AC,&9E,&1B,&A5,&78,&78,&2A,&BE,&28,&07,&9E,1620
120 DATA &1E,&C4,&2C,&2C,&DE,&29,&D0,&08,&26,&24,&AE,&78,&78,&C8,&06,&48,1559
130 DATA &01,&4A,&A0,&8E,&10,&C4,&2C,&18,&DE,&15,&D0,&02,&12,&FD,&A8,&C8,1749
140 DATA &0F,&FD,&0A,&FD,&40,&B5,&00,&9E,&53,&FD,&1A,&CD,&A2,&56,&E2,&E0,2199
150 DATA &E4,&6E,&90,&89,&05,&B5,&14,&AE,&79,&D1,&24,&9A,&B5,&00,&AE,&78,1994
160 DATA &63,&B5,&20,&AE,&79,&D1,&9A,&00,&00,&00,&00,&00,&00,&00,&00,&00,970
170 I=PEEK &7863*256:J=(PEEK &7863+1)*256:L=0
172 FOR A=0TO &C4:POKE J+A,PEEK (I+A):NEXT A
180 C=0:FOR I=ITO I+15:READ A:C=C+A:POKE I,A:NEXT I:L=L+10:READ A:IF C<>AGOTO
"ERR
190 IF A<>970GOTO 180
200 A=PEEK &7863:I=A*256:B=A/2:A=A+1
210 POKE I+&3C,A:POKE I+&ED,A
220 POKE I+&E6,B:POKE I+&F2,B
230 POKE I+&49,A-1:POKE I+&5B,A-1
240 POKE I+&91,A-1:POKE I+&AD,A-1
250 IF PEEK &E2B9=&38POKE &785B,A-1,&3B:POKE &79D4,85:END
255 POKE A*256+&8,32,ASC "I",ASC "N",ASC "I",0
260 POKE A*256+&22,0
270 POKE A*256+&3C,0
280 POKE A*256+&56,1,&F1,&8A,&28,&F1,&6F,&26,&37,&38,&36,&33,&2D,&31,&29,&2A,&
32
290 POKE A*256+&66,&35,&36,&2B,&26,&45,&43,&40,&0
300 CALL (A-1)*256+&EC:END
1000 "ERR"PRINT "ERREUR EN":L:END
```

LISTING 5

```
10 "A"RESTORE :BEEP ON
20 DATA 142,300,126,250,142,300,169,600,142,300,126,250,142,300,162,600,93,50
0,93
30 DATA 300,79,400,93,250,112,200,105,60[illegible]2,800,93,500,93,250,112,600,105,40
0,105
40 DATA 300,142,800,105,400,142,350,169,20[illegible]142,350,160,300,192,250,142,600,1
42
50 DATA 600,126,350,126,300,105,400,112,300,126,300,142,300,126,300,142,300,1
69,600
60 DATA 93,300,93,300,79,300,93,300,112,300,105,600,82,600,126,300,126,200,10
5,200
70 DATA 112,200,126,200,142,300,126,200,142,200,169,600,105,300,142,200,169,2
00
80 DATA 142,300,160,200,192,200,215,1000
90 ON ERROR GOTO 110:READ A,B:BEEP 1,A,B:GOTO 90
110 END
120 "B"RESTORE :BEEP OFF
130 ON ERROR GOTO 140:READ A,B:PLAY 1,A,B:GOTO 130
140 END

STILLE NACHT    AUTEUR:FREDERIC GUYON
```

MYSTERE

Lors de la mise au point du programme PC BLUESM, il est apparu un phénomène assez étrange.
Alors que le temps d'exécution du programme en langage machine prend exactement la même durée qu'il soit en ROM (adresse > &8000) ou en RAM normalement pour le basic (adresse < &7000), son temps d'exécution est doublé s'il se trouve dans la MEV système (adresse > &7000 et < &7C00)

EXEMPLE :

Mise à zéro de la zone [&8000-&FFFF] 255 fois

RAM ou ROM 110 secondes
RAM système 207 secondes

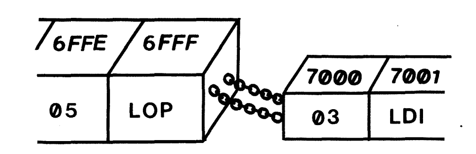
Il semblerait que ce soit lié à un temps d'accès différent des RAM.

Notez que l'on peut faire chevaucher un programme entre ces deux zones, et que la durée sera un panachage entre la durée en RAM et en RAM système. Les instructions peuvent même avoir leur code en RAM et leur paramètre en RAM système, donc en temps d'exécution encore différent.

EXEMPLE :

La moitié du programme précédent en RAM et en RAM système 156 secondes.

Marc GIRONDOT



2 types de RAM

JAQUETTE D'OR

Non, LE SHARPENTIER ne se convertit pas dans les pronostics hippiques, par contre ce programme vous permettra de rendre attrayante votre « CASSETTOTHEQUE » grâce à la création de jaquettes du plus bel effet.

Après le chargement faîtes :
RUN

Répondre au nombre de répétition qui correspond au nombre de fois où le texte sera écrit.
Répondre 1 si vos styles sont neufs et plus si ils sont légèrement usagés. Il est inutile de répondre 10 000 s'ils sont secs, il vaut mieux en changer !
Donnez les deux lignes à écrire, la 1 étant celle

du haut. Il ne faut pas plus de 23 caractères par ligne.
La couleur demandée correspond à celle du graphisme à gauche de la tranche.

Les corps correspondent au type de lettres.

CORPS 1

1 gras	rouge + bleu
2 gras	bleu
3 vide vertical	bleu
4 vide horizontal	bleu
5 plein vertical	bleu
6 plein horizontal	bleu

CORPS2

bâton	noir
gras	noir
vide vertical	noir
vide horizontal	noir
plein vertical	noir
plein horizontal	noir

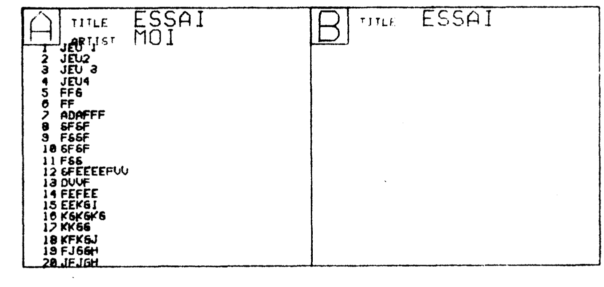
Répondre alors au type de graphisme par A ou B. On réservera par exemple le A aux programmes et le B à la musique.
Nous tenons à remercier, a cette occasion, notre SHARPENTIER traducteur KAMINO AWACHIKA qui nous fournit une aide précieuse pour l'adaptation de tous ces très bons programmes Japonnais.

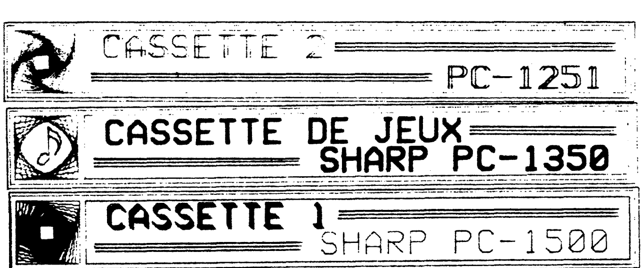
TITRES SUR CASSETTE 2

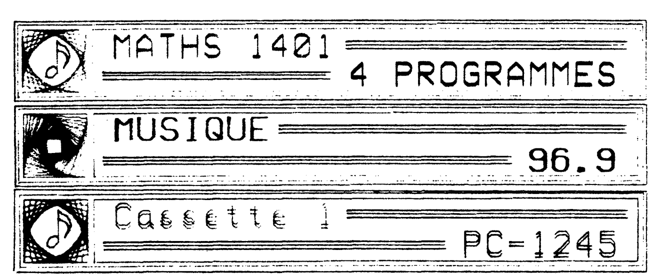
Après avoir décoré la tranche de vos cassettes, c'est la face avant qui sera imprimée cette fois.
DEFZ,répondre aux indications qui vous sont demandées et valider comme d'habitude par ENTER.
Vous êtes limités à dix titres par face, mais on peut aller jusqu'à 20 en changeant les tests des lignes 104 et 161.

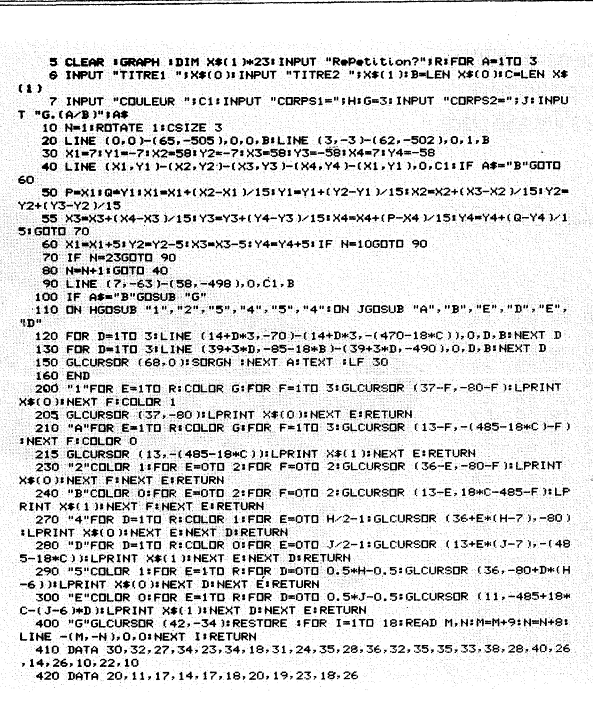
A TITRE ESSAI A
ARTISTE MOI MEME
1 JEU 1
2 JEU 2

B TITRE ESSAI B
ARTISTE ENCORE MOI MEME
1 JEU 3
2 JEU 4
3 JEU 5









```
5 CLEAR :GRAPH :DIM X$(1)*23:INPUT "RePetition?";R:FOR A=1TO 3
6 INPUT "TITRE1 ";X$(0):INPUT "TITRE2 ";X$(1):B=LEN X$(0):C=LEN X$
(1)
7 INPUT "COULEUR ";C1:INPUT "CORPS1=";H:G=3:INPUT "CORPS2=";J:INPU
T "G.(A/B)";A$
10 N=1:ROTATE 1:CSIZE 3
20 LINE (0,0)-(65,-505),0,0,B:LINE (3,-3)-(62,-502),0,1,B
30 X1=7:Y1=-7:X2=58:Y2=-7:X3=58:Y3=-58:X4=7:Y4=-58
40 LINE (X1,Y1)-(X2,Y2)-(X3,Y3)-(X4,Y4)-(X1,Y1),0,C1:IF A$="B"GOTO
60
50 P=X1:Q=Y1:X1=X1+(X2-X1)/15:Y1=Y1+(Y2-Y1)/15:X2=X2+(X3-X2)/15:Y2=
Y2+(Y3-Y2)/15
55 X3=X3+(X4-X3)/15:Y3=Y3+(Y4-Y3)/15:X4=X4+(P-X4)/15:Y4=Y4+(Q-Y4)/1
5:GOTO 70
60 X1=X1+5:Y2=Y2-5:X3=X3-5:Y4=Y4+5:IF N=10GOTO 90
70 IF N=23GOTO 90
80 N=N+1:GOTO 40
90 LINE (7,-63)-(58,-498),0,C1,B
100 IF A$="B"GOSUB "G"
110 ON HGOSUB "1","2","5","4","5","4":ON JGOSUB "A","B","E","D","E",
"D"
120 FOR D=1TO 3:LINE (14+D*3,-70)-(14+D*3,-(470-18*C)),0,D,B:NEXT D
130 FOR D=1TO 3:LINE (39+3*D,-85-18*B)-(39+3*D,-490),0,D,B:NEXT D
150 GLCURSOR (68,0):SORGN :NEXT A:TEXT :LF 30
160 END
200 "1"FOR E=1TO R:COLOR G:FOR F=1TO 3:GLCURSOR (37-F,-80-F):LPRINT
X$(0):NEXT F:COLOR 1
205 GLCURSOR (37,-80):LPRINT X$(0):NEXT E:RETURN
210 "A"FOR E=1TO R:COLOR G:FOR F=1TO 3:GLCURSOR (13-F,-(485-18*C)-F)
:NEXT F:COLOR 0
215 GLCURSOR (13,-(485-18*C)):LPRINT X$(1):NEXT E:RETURN
230 "2"COLOR 1:FOR E=0TO 2:FOR F=0TO 2:GLCURSOR (36-E,-80-F):LPRINT
X$(0):NEXT F:NEXT E:RETURN
240 "B"COLOR 0:FOR E=0TO 2:FOR F=0TO 2:GLCURSOR (13-E,18*C-485-F):LP
RINT X$(1):NEXT F:NEXT E:RETURN
270 "4"FOR D=1TO R:COLOR 1:FOR E=0TO H/2-1:GLCURSOR (36+E*(H-7),-80)
:LPRINT X$(0):NEXT E:NEXT D:RETURN
280 "D"FOR D=1TO R:COLOR 0:FOR E=0TO J/2-1:GLCURSOR (13+E*(J-7),-(48
5-18*C)):LPRINT X$(1):NEXT E:NEXT D:RETURN
290 "5"COLOR 1:FOR E=1TO R:FOR D=0TO 0.5*H-0.5:GLCURSOR (36,-80+D*(H
-6)):LPRINT X$(0):NEXT D:NEXT E:RETURN
300 "E"COLOR 0:FOR E=1TO R:FOR D=0TO 0.5*J-0.5:GLCURSOR (11,-485+18*
C-(J-6)*D):LPRINT X$(1):NEXT D:NEXT E:RETURN
400 "G"GLCURSOR (42,-34):RESTORE :FOR I=1TO 18:READ M,N:M=M+9:N=N+8:
LINE -(M,-N),0,0:NEXT I:RETURN
410 DATA 30,32,27,34,23,34,18,31,24,35,28,36,32,35,35,33,38,28,40,26
,14,26,10,22,10
420 DATA 20,11,17,14,17,18,20,19,23,18,26
```

```
10 "Z":GRAPH :CLEAR
11 GLCURSOR (0,0):SORGN
12 DIM A$(1)*22
13 DIM H$(1)*80
20 A=210:B=460:NA=0:U=0
21 DIM F$(1)*60
22 DIM G$(2)*30
26 LINE (-5,20)-(A+20,-B-20),5,2,B
30 LINE (0,0)-(A,-B),0,0,B
60 LINE (A-30,0)-(A,-30),0,0,B
70 COLOR 1
71 CSIZE 4:ROTATE 1
72 GLCURSOR (A-26.5,-7.5)
75 LPRINT "A"
76 COLOR 0
78 LINE (0,-B/2)-(A,-B/2)
80 LINE (A-30,-B/2)-(A,-B/2-30),0,0,B
85 COLOR 3:GLCURSOR (A-26.5,-B/2-7.5)
90 LPRINT "B"
91 L=0
94 GOSUB 4000
95 INPUT "TITRE DE LA FACE A?";H$(1)
96 U=LEN H$(1):IF U=0THEN 103
98 C=1:GOSUB 1000
99 GOSUB 4000
100 INPUT "ARTISTE DE LA FACE A?";H$(1)
101 C=2:GOSUB 1010
102 GOSUB 4000
103 INPUT "NOMBRE SUR FACE A?";NA
104 IF NA>10THEN GOSUB 4030:GOTO 103
105 IF NA=0THEN 107
106 GOSUB 2000
107 L=B/2:GOSUB 4000
108 INPUT "TITRE DE LA FACE B?";H$(1)
109 U=LEN H$(1):IF U=0THEN 160
110 C=3:GOSUB 1000
111 GOSUB 4000
112 INPUT "ARTISTE DE LA FACE B?";H$(1)
120 C=4:GOSUB 1010
159 GOSUB 4000
160 INPUT "NOMBRE SUR FACE B?";NA
161 IF NA>10THEN GOSUB 4030:GOTO 160
165 IF NA=0THEN GOTO 180
```

```
170 GOSUB 2000
180 GLCURSOR (0,-600):SORGN
184 GOSUB 4000
185 PAUSE "FIN?-(O/N)"
190 Z$=INKEY$ :IF Z$="O"THEN 500
191 IF Z$="N"THEN 26
195 GOTO 190
500 END
1000 C$="TITRE":K=1:GOTO 1020
1010 C$="ARTISTE":K=2
1020 U=LEN H$(1)
1030 IF U<=22THEN 1050
1031 PAUSE "DEPASSEMENT"
1032 GOSUB 4030
1040 ON CGOTO 95,100,108,112
1050 A$(1)=H$(1)
1055 CSIZE 1:COLOR 2:ROTATE 1
1060 GLCURSOR (A-15*K,-(L+40))
1070 LPRINT C$
1090 IF U<12THEN 1100
1095 CSIZE 1:COLOR 0:ROTATE 1:GOTO 1110
1100 CSIZE 2:COLOR 0:ROTATE 1
1110 GLCURSOR (A-15*K,-(L+90))
1150 LPRINT A$(1)
1180 H$(1)=""
1190 A$(1)=""
1191 U=0
1200 RETURN
2000 IF NA<6THEN GOTO 2500
2010 CSIZE 1:COLOR 0:ROTATE 1:J=1
2050 K=(180-NA*15*J)/(NA*2+1)
2100 FOR I=1TO NA
2110 GOSUB 4010
2120 INPUT "TITRE?";H$(1)
2130 F=LEN H$(1)
```

```
2151 GOTO 2120
2190 F$(1)=H$(1)
2200 GOSUB 3000
2210 GLCURSOR (A-(30+(7.5*J+K)*
     (2*I-1)),-(L+10))
2220 LPRINT I
2225 GLCURSOR (A-(30+(7.5*J+K)*
     (2*I-1)),-(L+31.5*J-(J-1)*15))
2230 LPRINT G$(1)
2240 GLCURSOR (A-(30+(7.5*J+K)*
     (2*I)),-(L+39*J-(J-1)*15))
2250 LPRINT G$(2)
2300 NEXT I
2310 NA=0
2315 H$(1)=""
2320 RETURN
2500 CSIZE 2:COLOR 0:ROTATE 1:J=2
2550 GOTO 2050
3000 G$(1)=LEFT$ (F$(1),30/J)
3010 G$(2)=MID$ (F$(1),30/J+1,30/J)
3030 RETURN
4000 BEEP 1,70,400
4005 RETURN
4010 BEEP 1,70,50
4020 RETURN
4030 BEEP 1,255,200
4040 RETURN
```

NEW & SANS NEW

Vous connaissez certainement cette instruction NEW suivie d'un paramètre, qui vous permet de déplacer la zone de programme BASIC.
Il y a au début de votre mémoire vive une zone de &C5 octets utilisée pour mémoriser le contenu des touches réserves. Puis se trouve la zone des programmes BASIC, mais entre deux zones il est possible d'insérer d'autres programmes en langage machine. On décale alors le début du BASIC par l'instruction NEW suivi de l'adresse où se situera le début du BASIC.

COMMENT CALCULER CETTE ADRESSE ?

Le début de votre mémoire vive est en PEEK &7863*256. Appelons &NN le poids fort de cette adresse. Le début de la RAM est donc en &NN00. A partir de &NNC5, on peut donc mettre le BASIC et le langage machine.
Si on fait NEW & MMMM, on définit le début du BASIC en &MMMM, le langage machine sera donc de &NNC5 à &MMMM−1. Pour entrer un programme de L octets à partir de &NNC5 on fera NEW&NNC6+L. Le problème à ce moment c'est que le programme BASIC contenu dans le PC sera effacé.
La routine proposée remédie à cet inconvénient qui divient vite insupportable.
Entrez le programme proposé où vous voulez, je vous laisse choisir l'endroit et le NEW à faire en guise d'exercice.
Mettre dans une variable par exemple A, l'adresse où doit se faire le NEW, et faire : CALL (adresse premier octet), variable.
Le début du BASIC est décalé, mais sans faire un NEW sur le BASIC. A noter que si la variable vaut 0, l'effet est le même qu'un NEW 0, mais toujours sans effacer le programme.
Bien sûr, le programme teste s'il y a la place sinon, il affiche ERROR 25.

Marc GIRONDOT

```
2A00:FD 5A 5C 00 89 02 5E 00  C6
2A08:89 06 A5 78 63 18 5A C5  78
2A10:CC 67 FD 88 FD 6A CC 65  8A
2A18:24 Fb 00 2A A4 80 28 F9  DO
2A20:14 22 0A 94 A2 08 FD 88  4D
2A28:A7 78 64 83 0A 94 A7 78  15
2A30:63 89 0F 5E C5 83 0b FD  03
2A38:0A FD 0A FD 0A FD 1A 68  F9
2A40:19 E0 A5 78 65 A7 78 67  6b
2A48:89 0D A5 78 66 A7 78 68  12
2A50:89 05 FD 0A FD 0A 9A 94  44
2A58:A7 78 65 89 0b 14 A7 78  CD
2A60:66 89 05 FD 0A FD 0A 9A  26
2A68:FD 0A FD 88 CA 67 CC 69  84
2A70:04 F9 12 2A 84 92 28 CC  DD
2A78:65 FD 88 24 Fb 00 0A A4  59
2A80:80 08 CA 69 FD 18 CA 65  A9
2A88:FD 0A 94 86 81 16 89 04  F7
2A90:14 06 81 10 FD 1A FD 2A  A3
2A98:46 67 53 84 A6 99 06 04  8F
2AA0:26 99 0A 9A FD 2A FD 2A  7b
2AA8:64 F5 A4 86 99 05 24 06  1D
2Ab0:99 09 9A 00 00 00 00 00  16
```



GESTION DU CLAVIER 3

Voici donc la partie consacrée à l'exploitation des connaissances acquises dans les deux précédents articles. Nous allons faire deux routines qui améliorent la gestion du clavier, puis, en étudiant les résultats de l'une des deux, nous allons expliquer les erreurs dans la lecture du clavier.

La première routine va servir à savoir si une touche du clavier est appuyée. Il va falloir qu'elle soit accessible facilement par le BASIC. On va donc donner un code ASCII au programme en LM, qui recherchera dans les tables de correspondance &FE80 - &FEBF et &FEC0 - &FEFF s'il la trouve, et la testera. La possibilité de passer des paramètres à un CALL sera exploitée ici pour passer le code ASCII de la touche à tester.

EXPLICATIONS SOMMAIRES *LISTE 6*

Le programme est relogeable vous pourrez l'entrer où bon vous semble. Par exemple en &40C6, vous ferez alors POKE &40C6, &FD, &36, &89, &03... Pour l'utiliser on a deux solutions :

1) On met dans une variable numérique le code ASCII de la touche à tester. On fait CALL [1ere adresse, ici &3000], [variable numérique] au retour, si la variable numérique vaut 0, c'est que la touche n'est pas appuyée, si c'est 1, c'est qu'elle est appuyée.

2) On met dans une variable alphanumérique le caractère correspondant à la touche à tester. On fait CALL [1ere adresse, ici &3000], [variable alphanumérique] au retour, la variable alphanumérique contient 0 (code ASCII &30) si la touche n'est pas appuyée, le code ASCII &31 soit 1 si elle est appuyée. Le programme marche même si vous appuyez sur plusieurs touches ensemble, et aussi si vous demandez les codes ASCII shiftés.

&3000 - &3008 Lecture du code ASCII demandé dans A
Si CALL [ ],
[variable alpha] A = lg max var
X = premier octet
Si CALL [ ],
[variable numérique]A = 00
X = valeur variable
&3009 - &3027 Recherche du code ASCII dans la table
x Pointe dans la table
Yh Pointe les PA
Yl pointe les IN
&3029 - &302D Pas des touches
- soit inexistante
- soit non appuyée
&302F - &303A Teste le clavier avec PA en entrée et IN en sortie
&303C - &303E La touche est appuyée
&3040 - &3053 Conversion en code ASCII si variable alpha
&3055 - &3056 Le SEC indique le chargement des paramètres dans la variable et retour
variable alpha : chaine de longueur A commençant en X
variable numérique : le nombre X

La seconde routine va plus ressembler à l'INKEY$ normal, mais en renvoyant toutes les touches appuyées. On va utiliser le CALL suivi d'une variable alphanumérique, non pour passer des paramètres, mais pour indiquer dans quelle variable on veut la réponse. Si on met une variable numérique, on aura une ERROR 7. Nous allons imiter la routine en &E42C. Puisque nous avons vu l'astuce des codes shiftés, nous allons l'exploiter : Si au lancement, la variable alpha est vide, ce sont les valeurs normales, mais s'il y a quelque chose dedans, on reverra les touches SHIFTEES.

EXPLICATIONS SOMMAIRES : *LISTE 7*

Placée en mémoire comme sur le listing, la routine se place immédiatement après la première, mais elle est relogeable. La syntaxe est : CALL [adresse premier octet, ici &3057], [variable alphanumérique]
Au retour : Si la variable était vide, on a dans celle-ci, les caractères que l'on obtiendrait avec une pression sur la touche, sinon on obtient ces mêmes touches, mais SHIFTEES.

&3057 - &3060 Si la variable est numérique ERROR 7
&3061 - &3063 Stockage des touches en &7B10 Tampon Alpha
&306F - &306D Détermination SHIFTEE ou non dans XL
&306F - &3073 Préparation des registres :
Xh = poids fort de la table de correspondance
A = niveau d'exploitation des PA
on sauve A car ce dernier est modifié par E41A
&3074 - &3077 Z = 1 si pas de touche pour ce PA
&3079 - &3080 PA suivant, on avance de 8 octets dans la table de correspondance.
&3082 - &308C X = &7B10
A = nb de touches et retour
&308D - &3095 Cherche lesquels des 8 bits de A sont allumés
&3097 - &309A On charge dans le tampon Alpha la touche

LES ERREURS :

Ecrivez ce petit programme :

```
10 : A$ = "  " : CALL &3057, A$ : WAIT 0 :
PRINT A$ : GOTO 10
RUN
```

appuyez sur 2. et – , vous verrez
52 – . Que vient faire ici ce 5 ?
Le programme proposé se planterait-il ?
Essayez avec INKEY$

```
10 : A$ = INKEY$ : WAIT 0 : PRINT A$ :
GOTO 10
RUN
```

appuyez sur 2 . et – , vous verrez
5 !! surprise !!

Regardons ce qui se passe au niveau électrique. A chaque intersection, si une touche est enfoncée, le courant va perpendiculairement vers la gauche pour sotir au niveau des IN : voir figure 1. Mais une partie va vers la droite et une autre continue.

● touche appuyée
— courant arrivant par PA
... courant sortant

Ce sont ces courants « parasites » qui produisent cet effet.
On se rappelle que INKEY$ teste les touches par ordre IN décroissant, PA croissant, ce qui correspond à celà sur la figure 1 :

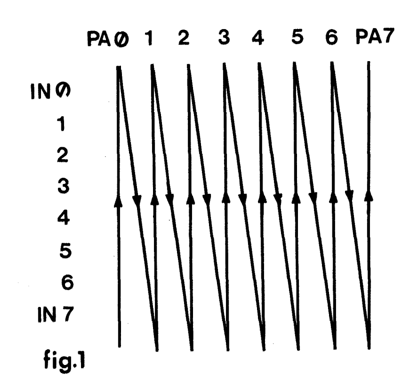
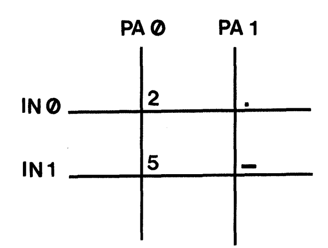


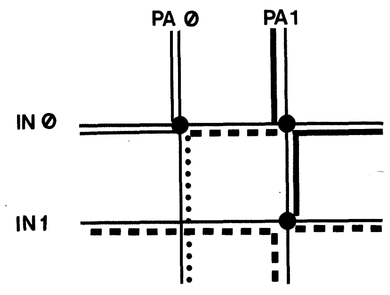
fig.1

Voyons ce qui se passe lorsque 2. et – sont appuyés :



Pour commencer, il envoie un courant en PA 0. La touche 2 est appuyée, une partie sort en INO, mais une autre partie va vers la touche ., qui étant appuyée ressort du courant en PA1 et vers la touche – qui étant aussi appuyée envoie du courant entre autre vers la touche 5 donc vers IN1
INO et IN1 sont allumés INKEY$ traite en commençant par les IN les plus forts donc pour lui IN1 est prioritaire, 5 est le résultat !

C'est donc le courant parasite 3 qui donne la touche 5 comme étant appuyée.



— conducteur
● touche appuyée
═ courant « principal »
⁞ courant parasite 1
▬ courant parasite 2
╍ courant parasite 3

Nous avons vu avec ces 3 articles, le principal du clavier mais il en reste beaucoup à voir, rassurez-vous !

Marc GIRONDOT

LISTE6

```
3000:PSH  A          3000:FDC8
3002:CPA  VL         3002:36
3003:BZR  3008       3003:8903
3005:LDA  XL         3005:04
3006:BCH  3009       3006:8E01
3008:LDA  (X)        3008:05
3009:LDI  YH,01      3009:5801
300B:LDI  YL,80      300B:5A80
300D:LDI  XH,FE      300D:48FE
300F:LDI  XL,80      300F:4A80
3011:CPA  (X)        3011:07
3012:BZS  302F       3012:8B1B
3014:PSH  A          3014:FDC8
3016:LDA  YL         3016:14
3017:SHR             3017:D5
3018:STA  YL         3018:1A
3019:BCR  3024       3019:8109
301B:LDI  YL,80      301B:5A80
301D:LDA  YH         301D:94
301E:SHL             301E:D9
301F:STA  YH         301F:18
3020:BCR  3024       3020:8102
3022:LDI  YH,01      3022:5801
3024:POP  A          3024:FD8A
3026:INC  XL         3026:40
3027:BCR  3011       3027:9118
3029:LDI  XH,00      3029:4800
302B:LDI  XL,00      302B:4A00
302D:BCH  3040       302D:8E11
302F:LDA  YL         302F:14
3030:STA  (734E)     3030:AE734E
3033:LDA  YH         3033:94
3034:SJP  E41A       3034:BEE41A
3037:BIT  (734E)     3037:AF734E
303A:BZS  3029       303A:9B13
303C:LDI  XH,00      303C:4800
303E:LDI  XL,01      303E:4A01
3040:POP  A          3040:FD8A
3042:CPA  VL         3042:36
3043:BZS  3053       3043:8B0E
3045:CPI  XL,00      3045:4E00
3047:BZR  304F       3047:8906
3049:LDI  XH,C8      3049:48C8
304B:LDI  XL,0A      304B:4A0A
304D:BCH  3053       304D:8E04
304F:LDI  XH,CA      304F:48CA
3051:LDI  XL,18      3051:4A18
3053:LDI  A,01       3053:B501
3055:SEC             3055:FB
3056:RTN             3056:9A
```

LISTE 7

```
3057:CPA  VL         3057:36
3058:BZR  3061       3058:8907
305A:POP  Y          305A:FD1A
305C:POP  Y          305C:FD1A
305E:LDI  UH,07      305E:6807
3060:ERH             3060:E0
3061:LDI  YH,7B      3061:587B
3063:LDI  YL,10      3063:5A10
3065:LDA  (X)        3065:05
3066:CPA  VL         3066:36
3067:BZR  306D       3067:8904
3069:LDI  XL,80      3069:4A80
306B:BCH  306F       306B:8E02
306D:LDI  XL,C0      306D:4AC0
306F:LDI  A,01       306F:B501
3071:LDI  XH,FE      3071:48FE
3073:STA  UH         3073:28
3074:SJP  E41A       3074:BEE41A
3077:BZR  308D       3077:8914
3079:LDI  A,08       3079:B508
307B:REC             307B:F9
307C:ADC  XL         307C:02
307D:STA  XL         307D:0A
307E:LDA  UH         307E:A4
307F:SHL             307F:D9
3080:BCR  3073       3080:910F
3082:LDI  XH,7B      3082:487B
3084:LDI  XL,10      3084:4A10
3086:SEC             3086:FB
3087:LDA  YL         3087:14
3088:LDI  YL,10      3088:5A10
308A:SBC  YL         308A:10
308B:SEC             308B:FB
308C:RTN             308C:9A
308D:LDI  UL,07      308D:6A07
308F:SHL             308F:D9
3090:BCS  3097       3090:8305
3092:INC  XL         3092:40
3093:LOP  308F       3093:8806
3095:BCH  307E       3095:9E19
3097:TIN             3097:F5
3098:LOP  308F       3098:880B
309A:BCH  307E       309A:9E1E
```

BOGUES EN ROM

Au fil des mois, vos appels, votre courrier et nos recherches nous ont permis de déceler, dans la ROM BASIC du PC-1500, un certain nombre d'erreurs conduisant à une exécution farfelue ou à un « plantage » de la machine.
Nous vous les livrons ici, dans l'ordre chronologique de nos découvertes.

BOGUES EN ROM AU NIVEAU DU BASIC

ANCIENNES ROM

* Le PC-1500 n'accepte pas d'étiquette seule sur une ligne :
exemple : 10 : « ESSAI »
RUN ERROR
* NOT NOT 1 donne un résultat farfelu
* Répondre une chaine vide à un INPUT provoque une erreur 0
* Un using suivi d'une variable numérique bloque le PC-1500

NOUVELLES ET ANCIENNES ROM :

* CSIZEO est accepté et conduit à des caractères de taille 256
* + ou – devant STR$ provoque l'erreur 131
* Les arguments de DATA, s'ils sont séparés par « ; » provoquent une erreur 0 à la lecture par READ.
* AREAD variable . . . . : ou &0D
ARUN . . . . : ou &0D

Tout ce qui se trouve entre variable et : ou &0D pour AREAD et ARUN et : ou &0D pour ce dernier, est ignoré.

* BEEP n,f,d, avec 0<n<1 provoque un BEEP perpétuel
* PRINT # – 2, « NOM » ; VAR ou plus simplement PRINT # – 2,
ou INPUT # – 2, « NOM » ; VAR ou plus simplement INPUT # – 2, provoque un SPJ &2C4 soit en RAM si 16Ko
Retour du SPJ par &E2 mais ERROR 1 sinon &CD &42 provoque un retour au moniteur
Exemple : POKE &2C4, &BE,&E6,&69,&CD,&42 ENTER
PRINT # –2 ENTER provoque un BEEP 1
* ERROR 31 se produit en essayant de lister un programme qui a l'adresse de fin incompatible avec la version mémoire et ce par l'intermédiaire de la RS-232C ou //

DIFFERENCES ANCIENNES ET NOUVELLES ROM :

Les boucles : Lors d'une boucle FOR-NEXT, les anciennes ROM testent si la variable prend la valeur finale et incrémente, les nouvelles incrémente et teste ensuite.
Lecture des données : Si vous mettez une instruction derrière un DATA, dans toutes les ROM elle sera interprétée, mais une erreur se produira si vous essayez de lire cette donnée avec un READ et une ancienne ROM.

ESSAI LOGICIEL SUR PC-1500

Attention, un titre peut en cacher un autre... Sachez, en effet que malgré les apparences, SOFT MONITOR n'est pas un moniteur. Ce programme qui prend 9Ko de la mémoire de votre PC-1500 et nécessite donc un module 16Ko au moins, est en fait un assembleur. On peut même dires que c'est le digne successeur de PC MACRO DE RVS.

ESSAI DE SOFT MONITOR :

Comme tout bon assembleur vous travaillez sur un programme source, que vous écrivez ici avec l'éditeur du basic, c'est facile et très pratique d'emploi. Un nombre illimité de labels peut être utilisé. Ce qui est révolutionnaire à mon sens, c'est la technique de l'assemblage de ce programme.
Vous faites RUN, et chaque instruction s'assemble elle-même en s'exécutant. Pour vous montrer la puissance du système, voici comment faire pour créer, ce que l'on appelle des macros instructions, c'est-à-dire une suite d'instructions qui se répètent plusieurs fois :
200 GOSUB "MACRO"
...
1000 "MACRO" instructions de la macro
1100 RETURN
On peut utiliser tous les mots BASIC habituels pour faciliter l'assemblage. Nous avons vu la principale caractéristique de ce programme, mais il y en a bien d'autres, par exemple le "Sourceur".
Il s'appelle par la fonction PRGS (adresse début, adresse fin) et permet de remettre sous forme de programme source un routine déjà assemblée pour pouvoir ensuite y apporter des modifications et le réassembler.
Il y a de nombreuses autres fonctions, comme PROG qui permet de récupérer dans certains cas un programme après plantage, un désassembleur ou un listing de la table des labels.
Notez que ce programme permet d'intégrer le FAST LOAD de LOGI'STYCK, la procédure est donnée dans le mode d'emploi.
En conclusions, ce programme est idéal pour programmer de petites applications car dès que le source dépasse 5 à 6Ko, il faudra passer par plusieurs assemblages successifs en stockant les différents sources sur cassette. La notice est très bien faite, avec une partie initiation, un mode d'emploi et une liste des macro-instructions, le programme permettant de les assembler et de les désassembler. Vous y trouverez aussi des exemples, qui sont bien pratiques pour l'initiation.

Marc GIRONDOT

LIENS DE VMS ET VH

Il convient, tout d'abord, de rappeler ce que sont ces 2 notions :

* Vh est le poids fort d'un nouveau registre 16 bits nommé V (voir bulletin N° 10)

* VMJ est un super CALL qui permet de se brancher à une adresse précise en ROM ; il y en a 128 possibles. A chaque VMJ est associé un nombre qui indique son numéro, ce numéro est toujours pair, de &00 à &FE. La syntaxe est donc VMJ n où n est le numéro de la macro. En fin de ROM se trouve la table des adresses où le microprocesseur doit se brancher. Elle se trouve de &FF00 à &FFFF. Par exemple, la macro &34 lit les adresses &FF34 et &FF35 et se branche au résultat, ici : &DF23 (voir liste 1).

Toutes les adresses de cette table se dirigent en ROM, il n'est pas possible de voir ce que devient notre registre Vh juste après ce branchement.

On va user de témérité pour tenter de les détourner vers la RAM en mettant des numéros de MACROS impairs. Le microprocesseur se laissera berner et se branchera, comme si de rien n'était à une nouvelle adresse qu'il calculera de la même façon que précédemment.

FF30 : DC 16 D0 71 DF 23 DF 0F

La macro &30 se branche en DC16, la macro &31 se branchera en 16D0, or, cette adresse indique de la RAM avec un module 16K.

On dispose ainsi de 128 nouvelles macros qui sont, pour nombre d'entre-elles, en RAM. (voir liste 2)

Seule, la macro &FF nous pose un problème ; nous avons, en effet, 2 possibilités de branchement, suivant le mode de calcul d'adresse qu'emploie le microprocesseur. Utilisant forcément un registre interne, il incrémente donc sur 8 bits ou sur 16 bits.

Exemple :

N est un registre interne
n est le numéro de la macro = &FF

Incrément sur 16 bits.	Incrément sur 8 bits.
Nh = &FF	Nl = n = &FF
Nl = n = &FF	Nh = &FF
(N) : poids fort de l'adresse = 00	(N) : poids fort de l'adresse
INC N = &FFFF + 1 = &000	N1 = Nl + 1 = &FF + 1 = &000
(N) = Poids faible de l'adresse	(N) = Poids faible de l'adresse
Ici, le poids fort de l'adresse est 00 et le poids faible à l'adresse &0000 soit donc en RAM pour une 16K.	Ici, le poids fort est 00 et le poids faible à l'adresse à l'adresse &FF00 soit &DC

Malheureusement, l'incrément se fait sur 8 bits et cette macro &FF se branche donc en &00DC.

Une fois une macro envoyée en RAM, il suffit de lire Vh et, surprise, on obtient toujours &FF. Ceci nous conforte dans l'idée que V est le « résidu » d'un registre interne servant aux résultats intermédiaires, car, c'est exacement le résultat que l'on obtient dans Nh dans notre exemple. Vl, quant à lui, continue son existence monotone en étant bloqué à 0.

Il ne reste qu'à décoder les nouvelles macros se dirigeant en ROM pour vérifier si elles peuvent être utiles.

Marc Girondot Pascal Abrivard

LISTE 1

```
00 @:DCB7
02 @:DCB6
04 @:DCC6
06 @:D065
08 @:DDD9
0A @:DE5E
0C @:DE97
0E @:D461
10 @:DD2D
12 @:DF93
14 @:DFFA
16 @:DFF5
18 @:DF80
1A @:D2E6
1C @:FA89
1E @:FB2A
20 @:DF72
22 @:DF63
24 @:DEAF
26 @:DB87
28 @:DBB1
2A @:D03E
2C @:DCA6
2E @:D6C0
30 @:DC16
32 @:D071
34 @:DF23
36 @:DF0F
38 @:CE9F
3A @:CFFB
3C @:FA74
3E @:FB9D
40 @:C401
42 @:CA58
44 @:CA7A
46 @:CA80
48 @:DCF9
4A @:DCFD
4C @:DCE9
4E @:DCED
50 @:DA71
52 @:F663
54 @:F7B0
56 @:F73D
58 @:F084
5A @:E573
5C @:F61B
5E @:F7A7
60 @:F6B4
62 @:F88B
64 @:F7B5
66 @:F7B9
68 @:F715
6A @:F88F
6C @:F6FB
6E @:F080
70 @:F747
72 @:F7CE
74 @:F775
76 @:F75F
78 @:F72F
7A @:F7DD
7C @:F6E6
7E @:F01A
80 @:F707
82 @:F729
84 @:EF00
86 @:EB40
88 @:EDF6
8A @:ED5B
8C @:EE1F
```

```
8E @.EDB1
90 @.EDAB
92 @.ED00
94 @.EC5C
96 @.EA78
98 @.EC74
9A @.ECEB
9C @.ECB7
9E @.E4A0
A0 @.E234
A2 @.E655
A4 @.B888
A6 @.E451
A8 @.B88B
AA @.B88E
AC @.E88C
AE @.B891
B0 @.B894
B2 @.B897
B4 @.B89A
B6 @.B89D
B8 @.B8A0
BA @.E763
BC @.E4B7
BE @.E4A8
C0 @.DD08
C2 @.DCD4
C4 @.DCD5
C6 @.DD13
C8 @.DCC5
CA @.C001
CC @.DDC8
CE @.D45D
D0 @.D5F9
D2 @.DD1A
D4 @.DEE3
D6 @.DED1
D8 @.DF3B
DA @.C00E
DC @.DEBC
DE @.D6DF
E0 @.CD8B
E2 @.C400
E4 @.CD89
E6 @.F70D
E8 @.F661
EA @.F79C
EC @.F757
EE @.F7CC
F0 @.EFBA
F2 @.EE71
F4 @.DBBC
F6 @.DDB5
F8 @.E171
FA @.E22C
FC @.E22B
FE @.E000
```

LISTE 2

```
01 @.B7DC
03 @.B6DC
05 @.C6D0
07 @.65DD
09 @.D9DE
0B @.5EDE
0D @.97D4
0F @.61DD
11 @.2DDF
13 @.93DF
15 @.FADF
17 @.F5DF
19 @.80D2
1B @.E6FA
1D @.89FB
1F @.2ADF
21 @.72DF
23 @.63DE
25 @.AFDB
27 @.87DB
29 @.B1D0
2B @.3EDC
2D @.A6D6
2F @.C0DC
31 @.16D0
33 @.71DF
35 @.23DF
37 @.0FCE
39 @.9FCF
3B @.FBFA
3D @.74FB
3F @.9DC4
41 @.01CA
43 @.58CA
45 @.7ACA
47 @.80DC
49 @.F9DC
4B @.FDDC
4D @.E9DC
4F @.EDDA
51 @.71F6
53 @.63F7
55 @.B0F7
57 @.3DF0
59 @.84E5
5B @.73F6
5D @.1BF7
5F @.A7F6
61 @.B4F8
63 @.8BF7
65 @.B5F7
67 @.B9F7
69 @.15F8
6B @.8FF6
6D @.FBF0
6F @.80F7
71 @.47F7
73 @.CEF7
75 @.75F7
77 @.5FF7
79 @.2FF7
7B @.DDF6
7D @.E6F0
7F @.1AF7
81 @.07F7
83 @.29EF
85 @.00EB
87 @.40ED
89 @.F6ED
8B @.5BEE
8D @.1FED
8F @.B1ED
91 @.ABED
93 @.00EC
95 @.5CEA
97 @.78EC
99 @.74EC
9B @.EBEC
9D @.B7E4
9F @.A0E2
A1 @.34E6
A3 @.55B8
A5 @.88E4
A7 @.51B8
A9 @.8BB8
AB @.8EE8
AD @.8CB8
AF @.91B8
B1 @.94B8
B3 @.97B8
B5 @.9AB8
B7 @.9DB8
B9 @.A0F7
BB @.63E4
BD @.B7E4
BF @.A8DD
C1 @.08DC
C3 @.D4DC
C5 @.D5DD
C7 @.13DC
C9 @.C5C0
CB @.01DD
CD @.C8D4
CF @.5DD5
D1 @.F9DD
D3 @.1ADE
D5 @.E3DE
D7 @.D1DF
D9 @.3BC0
DB @.0EDE
DD @.BCD6
DF @.DFCD
E1 @.8BC4
E3 @.00CD
E5 @.89F7
E7 @.0DF6
E9 @.61F7
EB @.9CF7
ED @.57F7
EF @.CCEF
F1 @.BAEE
F3 @.71DB
F5 @.BCDD
F7 @.B5E1
F9 @.71E2
FB @.2CE2
FD @.2BE0
FF @.00DC
```

SOLEIL

Ce programme de 1500 pas de programme en Basic permet de suivre le mouvement du soleil. Il offre deux possibilités :
• Si l'heure est fixée, il donne la position du soleil (azimut et hauteur observée)
• Si la hauteur du soleil est fixée, on obtiendra sa direction (son azimut)et l'heure correspondante.
Vous pouvez ainsi savoir où et quand auront lieu les couchers de soleil, où que vous soyez, chez vous ou aux antipodes, à la mer ou à la montagne. En prime vous saurez chaque jour à quelle heure exacte il est midi !

LA POSITION DU SOLEIL ET DE L'OBSERVATEUR

Pour l'observateur terrestre, la position du soleil est généralement définie par deux angles qui sont l'azimut et la hauteur (cf. fig 1).
L'azimut est simplement l'angle que vous lirez sur votre boussole (compas pour les marins) lorsque vous visez dans la direction de l'astre. Evidemment ce n'est pas très pratique lorsque le soleil est très haut. Il faudra corriger cette lecture de la déclinaison magnétique, indiquée sur la plupart des cartes.
L'azimut varie de 0 à 360° et il est compté à partir du Nord dans le sens des aiguilles d'une montre : Nord = 0°, Est = 90°. Sud = 180°, Ouest = 270°. A quand les grades qui arrondiront les chiffres, simplifieront grandement les calculs et transformeront les miles marins en simples kilomètres ?
La hauteur est comptée à partir de l'horizon théorique, perpendiculaire au fil à plomb. Un astre au zénith (à notre verticale) a une hauteur de 90° et, s'il est à l'horizon(tale), sa hauteur est nulle. Elle peut même devenir négative si on observe le lever ou le coucher d'un astre depuis un relief.

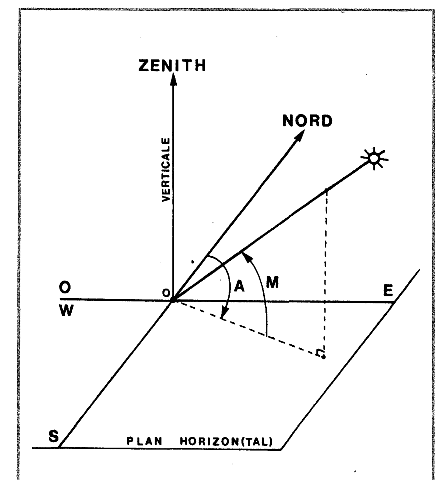


Figure 1
Un astre est localement repéré par deux angles (coordonnées sphériques) :
• l'azimut A : direction de sa projection sur le plan horizon, par rapport au nord.
• la hauteur M, hauteur mesurée au dessus du plan horizon(tal).
A un instant donné, il n'y a qu'un point sur terre d'où l'on puisse voir le soleil sous les angles A et M.

La position de l'observateur est définie en premier lieu par sa latitude et sa longitude. Les cartes marines sont graduées en degrés et minutes, avec pour origine des longitudes Greenwich : les cartes terrestres sont souvent graduées avec une origine prise sur la capitale du pays.

Le programme effectuera les corrections nécessaires pour ramener les coordonnées en degrés décimaux par rapport à Greenwich, qui est l'origine des angles horaires pour les éphémérides. Pour des pays autres que la France, on aura intérêt à modifier les lignes 6 et 9.

MODE OPERATOIRE POUR DETERMINER LA POSITION DU SOLEIL

Le calcul consiste tout d'abord à imaginer le soleil comme un point observé depuis le centre de la terre. C'est le domaine de la mécanique céleste qui fournit les coordonnées du soleil sur la carte du ciel. Il faut ensuite introduire deux corrections :
• le soleil n'est pas un point et il faut alors définir ce que l'on veut observer (le centre, le bord supérieur, le bord inférieur)
• vu de la surface de la terre, le soleil n'est pas tout à fait à la même position que vu du centre : il faut effectuer la correction de parallaxe (cf. fig. 2).

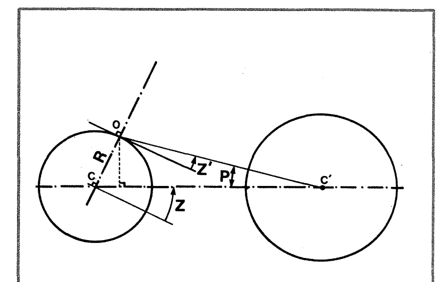


Figure 2
La hauteur Z' vue depuis la surface de la terre est toujours inférieure à la hauteur Z que l'on verrait du centre de la terre.
En considérant le triangle O C C' on obtient :
Z' = Z−P
Pour calculer P on utilise la relation approchée :
P = ATN (R★COS Z/D)
où R est le rayon de la terre (109 fois plus petit que celuis du soleil(et D la distance entre les deux astres).

Connaissant alors la position de l'observateur sur la terre (latitude et longitude), la trigonométrie sphérique permet de calculer exactement l'azimut et une hauteur de l'astre qui serait exacte si la terre était dépourvue d'atmosphère.
En effet à cause de la présence de l'air (cf. fig. 3), on ne voit jamais les astres là où ils sont, sauf au zénith, mais toujours au-dessus de leur position que nous qualifierons de "réelle".

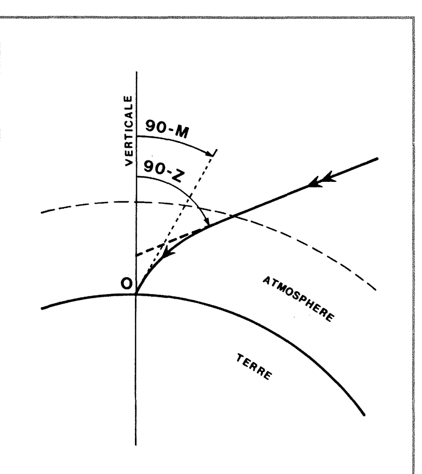


Figure 3
La présence de l'air courbe la trajectoire des rayons lumineux.
L'observateur situé au point O « voit » l'astre dans une direction plus proche de la verticale (hauteur plus élevée).
La différence M−Z se nomme la réfraction.

L'image que nous voyons n'est décalée ni à droite, ni à gauche ce qui signifie que l'azimut n'est pas affecté, mais la hauteur est relevée de la valeur de la réfraction que nous aborderons un peu plus loin.

CALCULS ASTRONOMIQUES

Le soleil ne tourne pas rond. Il peut présenter des retards ou des avances allant jusqu'à un quart d'heure sur le soleil fictif, au mouvement diurne uniforme, qui règle nos montres. On pourra consulter à ce sujet le programme « Cadran solaire », Revue Sharpentiers n° 12 - mai 1985.

Il faut donc calculer exactement sa position à partir des données astronomiques. Grosso-modo on donne un top départ (si l'on peut dire car il s'agit du printemps de l'an 2000 !), et on détermine la position exacte du soleil à partir de cette date. On utilise pour cela la loi de Képler et on projette sur les plans adéquats.

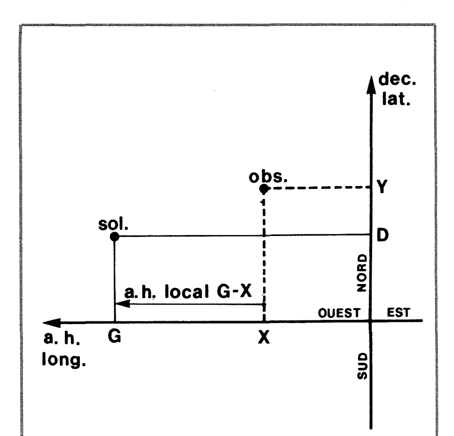


Figure 4
Un observateur terrestre est repéré par sa longitude X et sa latitude Y. Pour le soleil, il en est de même.
Si on imagine un axe allant au centre du soleil au centre de la terre, il perce la sphère terrestre en un point de longitude G (appelé angle horaire) et de lattitude D (appelée déclinaison).
Il faut évidemment imaginer cette figure sur une sphère.
Si G = X, le soleil passe au sud de l'observateur (ou au nord si D >Y).
Si cet observateur se place au point défini par G et D, le soleil est exactement à la verticale.

Les trois grandeurs intéressantes pour la suite du calcul sont :
- l'angle solaire qui est en fait la longitude du soleil (cf. fig. 4)
- la déclinaison, c'est-à-dire sa latitude
- le demi-diamètre solaire.

Il est tenu compte :
- des principales variations séculaires
- de la précession des équinoxes
- de la nutation
- du feston dû à la lune

mais l'influence des planètes est négligée faute de place.

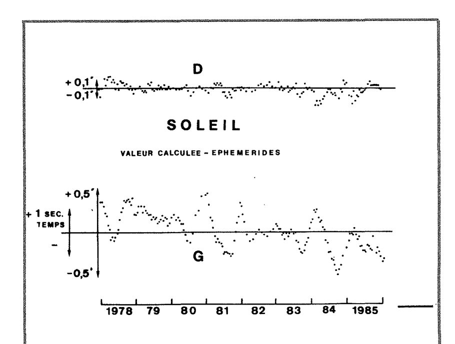


Figure 5
Comparaison entre les éphémérides officielles et le sous-programme de calcul de la position du soleil, de 1978 à 1985.
Pour la déclinaison D, l'erreur ne dépasse guère 0.1' d'angle, et, pour l'angle horaire G, l'erreur reste inférieure à 2 secondes de temps (0.5'). On peut aisément corriger l'écart sur G en agissant uniquement sur la variable Q ligne 101.

En fait si l'on compare les résultats du sous-programme astronomique avec les éphémérides (cf. fig. 5), on constate que :
- l'écart de la déclinaison du soleil est négligable,
- l'écart maximum sur son angle horaire est équivalent à 2 secondes de temps, ce qui est tout à fait honorable.

LINEARISATION JOURNALIERE

Les calculs astronomiques étant longs, nous avons linéarisé le mouvement du soleil sur chaque journée (cf. fig. 6).

Pour cela, la déclinaison et l'angle horaire sont calculés à deux instants de la journée, 6 h et 18 h, heures locales. Tous les calculs utilisent ensuite une interpolation sur ces points.

Attention : ligne 23 le chiffre 4 n'est pas une erreur ; il est rendu nécessaire par la redéfinition de G dans l'instruction précédente : on s'attend en effet à trouver un facteur 2 l'intervalle de temps étant égal à une demie journée.

Après détermination de l'heure de passage au méridien, les correspondances entre les angles et le temps deviennent très simples (cf. fig. 6).

On évite ainsi des calculs itératifs et le gain de temps est considérable.

Les erreurs introduites par la linéarisation sont tout à fait négligeables devant celles des calculs astronomiques dont nous venons de parler.

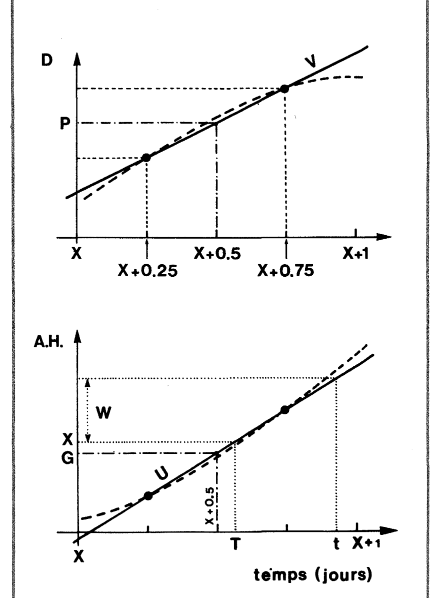


Figure 6
Pour faciliter les calculs, on peut approximer les variations de déclinaison et d'angle horaire par des droites (linéarisation).
Les erreurs introduites sont négligeables.
La déclinaison (cf. figure du dessus) et l'angle horaire sont calculés à 6 heures et 18 heures, heure locale, (X + 0.25 et X + 0.75). On ne garde en mémoire que les valeurs à 12 h locale P et G et les pentes des droites V et U.
La figure du dessous montre comment on relie l'angle horaire local W au temps écoulé à partir du midi local T.
Le temps T correspond au passage à la méridienne donc à un angle horaire égal à X.

N.B. : L'heure de passage au méridien correspond au moment où le soleil est le plus haut dans le ciel : c'est le vrai "midi" local, le milieu de la journée. Il n'a que rarement lieu à 12 heures à cause du mouvement non uniforme du soleil (voir « Calculs astronomiques ») et de l'écart en longitude.

REFRACTION

La réfraction (fig. 3) a un effet considérable. A l'horizon elle induit plus qu'un soleil d'écart (cf. fig. 7). Le bureau des longitudes donne un tableau des valeurs de cette réfraction pour des hauteurs observées de − 1° à + 90°.

De − 1° à 11°, nous avons représenté la réfraction à l'aide de deux polynômes, l'un de − 1° à 0°, l'autre de 0° à 11°.

Pour les hauteurs élevées (supérieures à 11°), nous avons opté pour la loi connue en 1/tg (hauteur).

Ainsi, de − 1° à 90°, nous rendons compte de la réfraction avec une erreur qui ne dépasse pas 3'' (0.05'). De plus le raccord, à 11°, entre la loi en tangeante et le polybôme est quasi parfait et les deux polynômes coïncident avec la valeur théorique de 36' à l'horizon (0°).

Toutes ces valeurs de réfraction sont données pour une température peu romantique de 0°C et une pression normale de 1013 mb. Or ces deux paramètres ont une très grande importance et leurs effets doivent être pris en compte. Nous dirons quelques mots sur les effets de la pression.

La réfraction est proportionnelle à la pression. Si les variations de pression dues aux anticyclones et dépressions sont faibles (quelques %), celles dues à l'altitude sont énormes : au sommet du Mont-Blanc (4807 m), la pression ne vaut plus que 55% de sa valeur au bord de la mer et la réfraction y est donc presque deux fois plus faible.

La procédure à suivre si on est en altitude est la suivante :

- soit on entre la pression locale lue en mb (ou bien on multiplie par : 1073/760 la pression lue en mm Hg) et on répond « zéro » à la question sur l'altitude.
- soit on entre la pression corrigée, c'est-à-dire ramenée au niveau de la mer, c'est la classique pression météorologique, et on entre l'altitude du lieu d'observation. le programme calculera alors la pression locale, en tenant compte de la loi de variation de pression en fonction de l'altitude.

Près de l'horizon, on pourra constater des écarts notables entre la réfraction-type et la réfracion réelle. La présence de masses nuageuses peut conduire à des anomalies de réfraction de plusieurs minutes d'angles ; chacun aura observé un soleil très distordu à l'horizon, beaucoup plus aplati par exemple que celui de la figure 7 calculé pour des conditions normales de réfraction.

LEVERS ET COUCHERS DE SOLEILS

A cause des anomalies de réfraction vers l'horizon, il est impossible de calculer le coucher du soleil à la seconde près.

L'heure que donne le programme correspond aux conditions moyennes de réfraction.

Dans les calendriers, les heures de lever et de coucher sont calculées pour un lieu donné (Paris pour la France), le soleil étant supposé se lever sur l'horizon théorique (hauteur = 0).

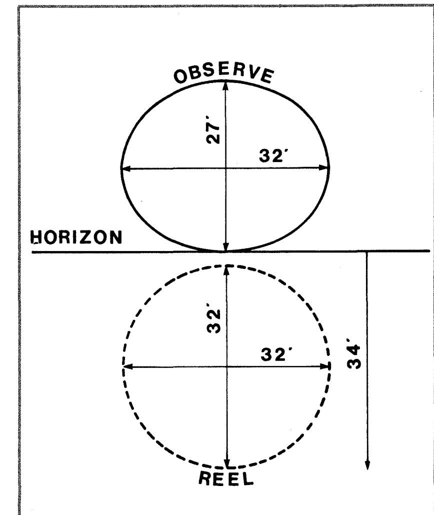


Figure 7
Début de coucher de soleil (ou fin de lever) sur l'horizon(tale), pour une température de 20°C et une pression normale.
Lorsque le soleil observé touche l'horizon, le soleil réel est déjà complètement couché.
On notera l'aplatissement du soleil, sa base étant plus réfractée que sa partie supérieure.

La réfraction est supposée constante durant toute l'année, généralement égale à 34', ce qui correspond à 20°C et 1013 mb, mais la valeur de 36' 36'', qui correspond aux conditions normales 0°C et 1013 mb est elle aussi utilisée. De plus, le rayon du soleil est souvent supposé constant et égal à 16' (il varie de 16.3' l'hiver à 15.7' l'été).

Le lever est souvent défini comme l'apparition du bord supérieur du soleil et le coucher sa disparition, mais certains considèrent son centre. Fort de toutes ces remarques, il ne faut donc pas s'inquiéter trop vite si l'on trouve une différence avec certaines publications.

L'HORIZON DU PROGRAMME

Si l'on dispose d'un instrument muni d'un horizon artificiel, ou si on utilise la visée directe plus la réflexion sur une surface horizontale (récipient d'eau ou de mercure), on mesure effectivement la hauteur de l'astre au-dessus de l'horizon théorique.

Pour les observations côtières, ou en mer, l'horizon visible, plus bas que l'horizon théorique est une base commode pour mesurer la hauteur des astres.

Lorsque le programme demande (ligne 81) ou donne (ligne 70) la hauteur observée, c'est par rapport à l'horizon visible, c'est donc exactement la mesure qui serait obtenue par un sectant.

Le fait d'être toujurs en dessus du niveau de la mer pour l'observation, entraîne une correction de dépression (cf. fig. 8), prise en compte par le programme : il suffit de répondre correctement à la question Hauteur/Horizon, en mètres (ligne 40).

Pour ceux qui veulent les hauteurs par rapport à l'horizon théorique, il suffit de répondre : 0 (zéro) à cette question.

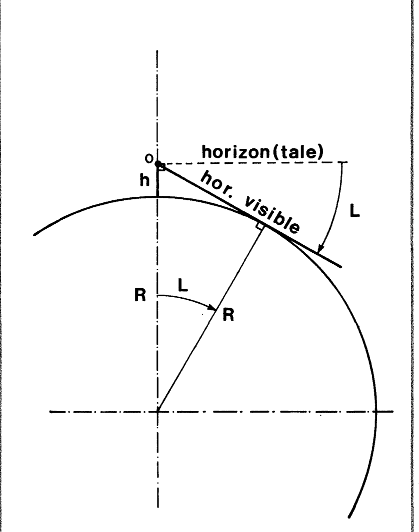


Figure 8
Pour un observateur situé à une hauteur h, l'horizon visible est en dessous de l'horizon théorique. L'angle L est défini par :
cos L = R/(R+h)
L s'appelle la dépression de l'horizon.

NB : La hauteur de l'horizon observée est égale à − L. Dans le programme, L est égale à : ation, entraîne une correction de dépression (cf. fig. 8), prise en compte par le programme : il suffit de répondre correctement à la question Hauteur/Horizon, en mètres (ligne 40).
Pour ceux qui veulent les hauteurs par rapport à l'horizon théorique, il suffit de répondre : 0 (zéro) à cette question.
NB : La hauteur de l'horizon observée est égale à − L. Dans le programme, L est égale à :

$L = \sqrt{(\text{hauteur(m)}/1140)}$

Si l'on se reporte à la figure 8, on arrivera aisément à la relation :

$\cos L = R / (h + R)$ où R est le rayon de la terre (6370 km).

En développant le cosinus en $1 - L^2/2$, on arriverait à la relation :

$L = \sqrt{(2h/R)}$ ou $L = \sqrt{(h/970)}$ avec h en mètres et L en degrés.

Pourquoi emploie-t-on la valeur 1140 et non

pas 970 dans le programme ? C'est encore une correction due à la réfraction car les rayons issus de l'horizon ne rejoignent pas l'observateur en ligne droite.
On peut simplifier ce genre de calculs en utilisant un rayon fictif de la terre R' égal à 7 500 km au lieu de 6 370 km. La réfraction est intégrée dans ce gonflement de la sphère terrestre.
Tant que nous y sommes, donnons la distance de l'horizon visible d'un hauteur h : c'est $\sqrt{}$ (2hR').

EVALUATION DE LA HAUTEUR DES RELIEFS

Si l'observateur se trouve à une altitude h1 et observe un coucher de soleil sur un relief d'altitude h2 situé à une distance d, la hauteur du soleil n'est pas égale à : arc tg ((h2 − h1)/d) à cause de la rotondité de la terre.
Un relief situé à une distance d « s'enfonce » de $d^2/2R'$ ou R' est le rayon fictif de 7 500 km dont nous venons de parler : cet enfoncement représente 1.5 km à 150 km de distance. Il est vrai qu'en altitude ce rayon R' se rapproche du rayon R de la terre, à cause de la diminution de la réfraction, mais ne nous compliquons pas...
Donc, la hauteur angulaire est égale à :
arc tg ((h2 − h1)/d − d/2R')
Si un observateur situé sur un relief veut calculer un lever ou un coucher de soleil, il lui faut savoir quel est le relief le plus « haut » au sens angulaire, et ensuite déterminer si ce relief est en dessus ou en dessous de l'horizon. Vous connaissez déjà la hauteur de l'horizon, c'est :
− L = − $\sqrt{}$(h1/1140)
La figure 9 représente une construction qui permet de tester rapidement quel est, de l'horizon et des différents reliefs, celui qui est à prendre en considération.

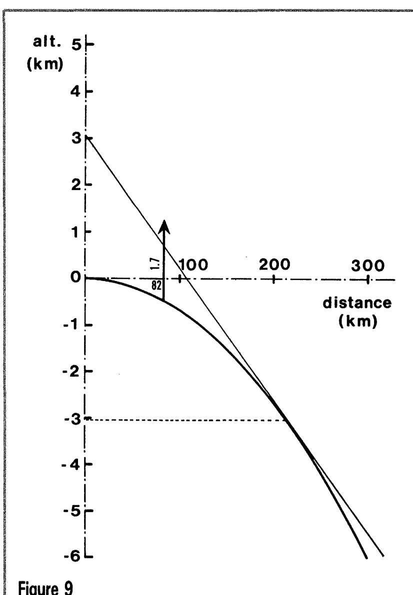


Figure 9
Détermination du relief prépondérant.
On peut localement assimiler un grand cercle terrestre à une parabole d'équation $-d^2/R'$ avec R' = 7 500 km, pour tenir compte de la réfraction, d étant la distance de l'observateur au point considéré.
Pour une altitude donnée, la ligne de l'horizon qui serait visible en l'absence de relief est tangente à la parabole. Pour tracer cette tangente, on peut utiliser une propriété de la parabole : le point de tangence a une altitude négative, symétrique par rapport à l'horizon. Il cache donc cet horizon et le soleil se couchera sur ce relief.

A L'INTENTION DES MARINS

Ce programme est parfaitement utilisable pour la navigation astronomique à l'aide du soleil, spécialement lorsqu'on emploie la méthode de la droite de hauteur (voir le programme « Navigation astronomique » dans la revue Sharpentier n° 10 - décembre 1984).
A la ligne 70, la variable A est l'azimut et M est la hauteur corrigée du soleil. L'intercept nécessaire à la correction de la droite de hauteur est égal à S − M si S est la valeur de la mesure du sextant. Toutes les corrections de demi-diamètre, de la réfraction, etc. sont déjà effectuées.

RECALAGE DE L'ANGLE HORAIRE

Si l'on estime que l'écart de deux secondes de temps sur l'angle horaire est trop important, il faut recaler la constante Q à la ligne 101.
Pour cela, faire tourner le programme pour une longitude nulle et comparer le temps de passage au méridien (H, ligne 27) avec les éphémérides.
Modifier la variable Q en conséquence :
+ 0.01 sur Q avance le temps de passage de 2.4 secondes.

TEMPS ECOULE ENTRE DEUX DATES - DONNEES ASTRONOMIQUES

La variable I, ligne 14, représente le nombre de jours à partir du premier janvier de l'an 2000 à 0 h (2000,0). I est donc négatif de nos jours. Nous connaissons donc le nombre de jours qui nous sépare de ce réveillon mémorable et, par différence, il est aisé de trouver le nombre de jours qui sépare deux dates. Notez que 1900 est bissextile alors que 2000 ne l'est pas.
Plusieurs données astronomiques intéressantes sont contenues dans le programme. Citons par exemple :
- l'anomalie excentrique (ligne 105) et vraie (ligne 106)
- le diamètre du soleil (ligne 108)
- l'ascension droite (ligne 109)
- la déclinaison D (ligne 110), l'angle horaire G (ligne 111) (attention G est exprimé en nombre de tours).

Il faut noter que le sous-programme de calcul des éphémérides du soleil ne nécessite que la connaissance du temps T compté en jours décimaux à partir de 2000,0. On peut très bien l'utiliser pour obtenir les éphémérides du soleil à un instant T quelconque.

A L'ATTENTION DES PROGRAMMEURS

Ce programme a été mis au point sur une SHARP PC 1211. Mais tel qu'il est présenté, il dépasse légèrement les 1424 pas de programme. Voici quelques conseils pour le compacter sur PC 1211 :
- supprimer les zéros inutiles
- supprimer les signes inutiles (ligne 14, écrire : INT.01Z, etc.)
- supprimer les parenthèses de fermeture
- supprimer les valeurs par défaut (ligne 3, 4...)
- supprimer les formats et les commentaires d'entrée
- être très succint dans les commentaires d'entrée
- G − INT G, ligne 25, est esthétique pour la suite (ligne 62) mais pas indispensable.
- supprimer *(1 − (A = − 1)) ligne 201 qui correspond à un cas d'école
- supprimer totalement la ligne 401, si vous ne vous intéressez pas aux hauteurs très négatives (< − 0.2°))

Faute de place, nous n'avons pas introduit la correction d'heure locale (1 heure ou 2 heures en Europe) ; c'est très simple, mais attention au signe. Dans le programme toutes les heures doivent être exprimées en temps universel (TU).

N.B. : La variable O (lettre O) n'intervient qu'aux lignes 100 et 105. Ne pas la confondre avec le chiffre 0.
Ligne 109, 5E − 6 signifie 0.000005.

MISE AU POINT DU PROGRAMME

Afin de faciliter la mise au point, nous donnons les valeurs des principales variables, pour un cas particulier (voir exemples).

Dans le cas où il y a plusieurs passages sur un sous-programme, nous donnons les valeurs correspondant au premier passage.

ligne	nom et valeur des variables
3	X = −5.0260 gr
4	Y = 50.9627 gr
5	Z = 1
6	W = 1
8	Y = 45.86643°
9	X = −0.01905722 tour
10	J = 24 , M = 7 , A = 1985
14	I = −5274 j
102	T = 5273.769057j , Z = −5353.083057j
103	F = −0.2044378
104	C = 23.44303°
105	A = −5201.15755°
106	A = −161.4643573°
107	S = 0.26277°
108	A = 123.503060°
109	G = −93.2735575 tour
110	D = 19.87926°
23	G = 0.9764393 tour, U =0.9999872 t/j
	P = 19.82648 ° , V = −0.211105 °/j
25	W = 0.0045035 t
27	H = 11h 39 mn 02.56 sec
40	M = 0
41	K = 3.05 km
42	B = 1028 mb, C = 14°
43	L = 0
44	N = 0.6605
50	F = −1
60	H = 19.2038 (19 h 20 mn 38 sec.
61	K = 0.3250526 j
62	W = 475.396175 °, D = 19.75787°
63	Z = −2.202823°'
65	Z = −1.94245°
68	C = 1.05243 ° (dernier passage)
70	M = −1° 14'50'' , A = 301° 42' 00''
50	F = −1
51	J = 1
80	B = 1
81	M = −1.1527 (−1° 15' 27''(
82	M = −1.2575°
83	Z = −2.21630°
302	W = 115.5090° (premier passage)
85	D = 19.7578°
302	W = 115.41883 ° (deuxième passage)
90	H = 19 h 20 mn 43.4 sec , a = 301° 43' 00''

EXEMPLES

1. Coucher de soleil le 24 juillet 1985 au refuge des Grands Mulets

Cet exemple est celui traité dans la mise au point du programme.

Longitude : −5.0260 gr/Paris
Latitude : 50.9627 gr
Altitude : 3050 m
Pression : 1028 mb
Température : 14° C

Le soleil se couche sur la chaîne du Jura, située à 82 km de distance, dont la crête avoisine 1 700 m d'altitude.

La rotondité de la terre entraîne un abaissement de :

$$\frac{(82)^2}{2\star 7500} = 0.45 \text{ km}$$

Dans ces conditions la hauteur du soleil au coucher est de :

DMS ATN ((1.7−0.45−3.05)/82) = 1° 15' 27''

L'heure du coucher, calculée à l'aide du programme est égale à :

19 h 20 mn 43 s T.U. (21 h 20 mn 43 s heure européenne).

Ce soir-là, l'horizon était très clair, et nous avons observé le coucher de soleil à :

19 h 20 mn 38 s T.U.

Trop beau pour être vrai ?

2. Lever de soleil, le 25 juillet 1985 sur la Grande Bosse (arête ouest du Mont-Blanc)

Longitude : −5.0201 gr/Paris
Latitude : 50.9288 gr
Altitude : 4 500 m
Pression : 1028 mb (météo)
Température : −2°C
Hauteur observée/horizon(tale) = −1° 07' 30''
Azimut = 58° 44' 17''

Faute de carte détaillée de la Suisse (et de l'Autriche ?), nous ne pouvons pas vérifier la valeur de la hauteur pour l'azimut donné.

A vous de jouer, si vous disposez du nécessaire, et, s'il vous plaît, faites part de vos observations à l'auteur, qui vous remercie par avance.

Gilbert VINCENT

2 Travail machine en degrés décimaux

3 COORDONNEES DE L'OBSERVATEUR
Longitude ? Latitude ?
en grades ou en degrés ?
origine Greenwich ou Paris ? 9

10 NOMBRE DE JOURS / 2000,0
Jour ? Mois ? Année ? 14

20 LINEARISATION DECLINAISON
ET ANGLE HORAIRE 23

25 PASSAGE AU MERIDIEN
27 Sortie h.mn.s TU

40 CONDITIONS D'OBSERVATION
Hauteur/Horizon ?
Altitude ?
Pression ? Température ? 44

100 *** SOUS-PROGRAMME EPHEMERIDES ****

100 Obliquité, excentricité, longit. périgée, temps périgée
101 Printemps 2000, Angle horaire pour P Année sidérale
102 Temps/2000,0 Temps/printemps 2000
103 Angle nutation, Précession
104 Obliquité écliptique
105 Anomalie excentrique (Képler)
106 Anomalie vraie
107 Rayon solaire
108 Ascension droite
109 Angle horaire en nb de tours avec feston lune (5E-6=0.000005)
110 Déclinaison
112

90 Sortie
Azimut ° ' "
Heure h mn s TU

50 BORD SUP/INF ?

70 Sortie
Azimut ° ' "
Hauteur ° ' "

51 On fixe l'HEURE ou la HAUTEUR?

60 HEURE ?
h mn s T.U

MATIN OU A.M ? 80

HAUTEUR ? 81
° ' "

61 Temps/Midi
Angle horaire
Hauteur astronomique

65 1/2 diamètre + parallaxe

mauvais

67 Test Précision

bon

68 Hauteur Observée

200 Sous-prog. AZIMUT

300 Sous-prog. ANGLE HORAIRE LOCAL

400 Sous-prog. REFRACTION

déclinaison affinée

85 déclinaison approchée

84 hauteur réelle/horizon

```
1    "SOLEIL"
2    R=360 : DEGREE
3    X = 00.0000 : INPUT "LONGITUDE (O+,E-) ?"; X
4    Y = 45.0000 : INPUT "LATITUDE  (N+,S-) ?"; Y
5    Z = 0 : INPUT "DEGRE = ENTER  , GRADE=1  "; Z
6    W = 0 : INPUT "GREENWICH=ENTER, PARIS=1  "; W
7    IF  Z=1 LET X = 0.9*X : Y = 0.9*Y : GOTO 9
8    X = DEG X : Y = DEG Y
9    X = (X -2.3372*W) / R

10   INPUT "JOUR ?"; J ,"MOIS ?"; M ,"ANNEE NNNN ?"; A
11   Z = A-2000
12   IF M > 2 THEN 14
13   Z = Z-1 : M = M+12
14   I = J + INT (30.6*(M+1)) - 63 + INT (365.25*Z)
        -INT (Z/100) + INT (INT(Z/100)/4)

20   H = X + 0.25 : GOSUB 100
21   U = G : V = D
22   H = H + 0.5  : GOSUB 100
23   G = (G+U)/2 : U = 4*(G-U) : P = (D+V)/2 : V = 2*(D-V)
25   W = X - G : W = W - INT W : W = W - (W>0.5)
26   T = X + 0.5 + W/U : T = T - INT T : H = DMS (24*T)
27   PRINT USING "###.#####" ; "MIDI="; H

40   M = 0 : INPUT "HAUTEUR / HORIZON (M) ?"; M
41   K = 0 : INPUT "ALTITUDE (KM) ?"; K
42   B = 1013 : C = 14 : INPUT "P (MB)?"; B ,"T (DC)?"; C
43   L = \/‾ (M/1140)
44   N = B*(1-K/42)^5 / 1013 / (1+C/260)

50   F = -1 : INPUT "BORD SUP=ENTER , CENTRE=0 ,INF=1 "; F
51   INPUT "HEURE = ENTER , HAUTEUR = 1 "; J : GOTO 80*J

60   INPUT "HEURE TU H.MNS ?"; H
61   K = DEG H /24 - X - 0.5 : K = K -INT K : K = K - (K>0.5)
62   W = R * (G + K*U - X) : D = P + K*V
63   Z = ASN ( COS D * COS Y * COS W + SIN D * SIN Y)
64   GOSUB 200
65   Z = Z - S * (F + COS Z / 109) : M = Z : E = 0
66   GOSUB 400
67   IF ABS (C-E) < 1E-4 THEN 69
68   M = Z + N*C : E = C : GOTO 66
69   M = DMS (M + L)

70   PRINT USING "####.####" ; "A=" ; A ; "M=" ; M
71   GOTO 50
80   B = -1 : INPUT "MATIN = ENTER , AM = 1 " ; B
81   INPUT "HAUTEUR OBSERVEE (D.MNS) ?"; M
82   M = DEG M - L : GOSUB 400
83   Z = M - N * C + S * (F + COS M / 109)
84   D = P : GOSUB 300
85   D = P + V * (W/R/U + T - X - 0.5) : GOSUB 300
86   GOSUB 200
87   H = T + W/R/U : H = H - INT H : H = DMS (24*H)
90   PRINT USING "####.####"; "A=" ; A ; "H=" ; H
91   GOTO 50
99   END
100  C = 23.4393 : E = 0.016710 : L = 77.060 : O = 3.016909
101  P = 79.3140 : Q = 291.1595 : B = 365.256363
102  T = I + H  : Z = T - P
103  K = 125 - 0.053*T : F = 0.014 * (Z/B - (SIN K-0.86)/3))
104  C = C + 0.0026 * (COS K - T/B/20)
105  FOR K = 1 TO 4 : A = (T-O)*R/B + E*R/2π* SIN A : NEXT K
106  A = 2* ATN (TAN (A/2) * \/‾ ((1+E)/(1-E)))
107  S = 0.267 * (1 + E*COS A)
108  K = A - L + F : A = ATN (TAN K * COS C)+ R/2*(ABS K>90)
109  G = H - P + Z/B + ( Q - A + F*COS C ) / R
        - 5E-6 * (SIN (R*(T-6)/29.53) - 0.11)
110  D = ASN (SIN K * SIN C)
111  RETURN

200  A = (COS D * SIN Y * COS W - SIN D * COS Y) / COS Z
201  A =  DMS (SGN SIN W * ACS A + R/2*(1-(A=-1)))
202  RETURN

300  W = (SIN Z - SIN D * SIN Y) / COS D / COS Y
301  IF ABS W > 1 : BEEP 5 : GOTO 50
302  W = B * ACS W
303  RETURN

400  IF M>11 LET C = 1 / TAN M / 61 : GOTO 403
401  IF M<0  LET C = 0.61 -0.23*M + M*M/10 : GOTO 403
402  C = 0.61 / (1 + 0.4*M + M*M/33*(1-M/22))
403  RETURN
```

L'EMPIRE CONTRE-ATTAQUE

On l'attendait depuis longtemps... Enfin le voilà ! Un jeu d'action en langage machine sur le PC 1251. De quoi s'agit-il ? Le jeu est librement inspiré de la célèbre bataille du début du film « L'empire contre-attaque ». Pour ceux qui ont vu le film et les autres, je rappelle brièvement la situation.

L'alliance, ennemie jurée de l'ineffable Empire, a établi un poste avancé sur la planète glacée d'Hoth. Mais l'affreux Dark Vador ne tarde bientôt pas à retrouver leur position, et lance l'armée avec votre PC 1251. Aux commandes d'un canon laser, vous devez détruire les Ambulos, véhicules tout terrain de l'Empire, qui attaquent la base. Il faut que vous les reteniez le plus longtemps possible, pour permettre aux restes des forces rebelles de se replier dans l'espace pour y continuer la lutte. Même, si vous le pouvez, détruire entièrement les forces impériales. Mais ne vous faites pas trop d'illusions, plus le temps passe, plus votre résistance diminue, plus l'ennemi avance vite. Je vous dirai entre nous, que je ne suis jamais arrivé à bout de l'armée impériale. Mais je n'ai jamais été très bon en jeu vidéo, même avec ceux que je crée...

QUELQUES DETAILS PRATIQUES

Il faut que vous laissiez approcher le moins possible les Ambulos de votre canon. Pour cela, il faut que vous tiriez le plus souvent possible, cela entrave la marche des Ambulos. Le gagnant sera celui qui aura permis le plus grand nombre de survivants.
Bonne chance !!! Et que la force soit avec vous...

Maintenant, redevenons sérieux et étudions ce programme de fond en comble. Je ne vais pas prétendre faire un cours de programmation mais, seulement expliquer le fonctionnement de certaines routines en langage machine, que j'ai essayé de rendre les plus universelles possible. Pour permettre une éventuelle réutilisation dans d'autres programmes.

PROGRAMME NUMERO 1 : LA ROUTINE MUSICALE

Ce programme est très simple à réutiliser. Il suffit que vous soyez un peu musicien et que vous compreniez la suite du chapitre. A chaque note correspond quatre paramètres.

1. Les deux premiers paramètres se lisent sur la figure 1, il s'agit de constantes de temps, qui imposent la fréquence de vibration du buzzer. Exemple :
Pour un fa1 correspondent les paramètres : 255, 220.
Pour un do # 2 correspondent les paramètres : 255, 10.
Pour un la2 correspondent les paramètres : 0, 150.
etc.

	FA #1	SOL #1	LA #1	DO #2	RE #2	FA #2	SOL #2	LA #2
	CF1 255	CF1 255	CF1 255	CF1 255	CF1 000	CF1 000	CF1 000	CF1 000
	CF2 200	CF2 125	CF2 075	CF2 010	CF2 235	CF2 180	CFE 160	CF2 140
	CDR 061	CDR 073	CDR 085	CRD 105	CRD 119	CDR 155	CDR 175	CDR 200

MI	FA	SOL	LA	SI	DO	RE	MI	FA	SOL	LA	SI	DO
CF1 255	CF1 255	CF1 255	CF1 255	CF1 255	CF1 255	CF1 000	CF1 000	CF1 000	CF1 000	CF1 000	CF1 000	CF1 000
CF2 255	CF2 220	CF2 150	CF2 100	CF2 050	CF2 020	CF2 250	CF2 220	CF2 200	CF2 170	CF2 150	CF2 125	CF2 110
CDR 055	CDR 059	CDR 069	CDR 079	CDR 091	CDR 102	CDR 112	CDR 127	CDR 140	CDR 165	CDR 187	CDR 224	CDR 255

CF1 = PARAMETTRE 1 ; CF2 = PARAMETTRE 2 ;
CDR = CONSTANTE DE RETARD

fig.1 CLAVIER DES VALEURS DE PARAMETRES

2. Le troisième paramètre correspond à une constante de rattrapage dont nous définirons l'utilité tout-à-l'heure. Cette constante est la troisième valeur que l'on peut lire sur la figure 1 : exemple :
Pour un sol1 correspond le paramètre : 69
Pour un ré # 2 correspond le paramètre : 119
Pour un do2 correspond le paramètre : 255

3. La quatrième valeur est la durée pendant laquelle on joue la note :
9 = ronde
5 = blanche
3 = noire
2 = croche
1 = double croche

Si vous voulez une vitesse supérieure à la double croche, vous laissez à 1 le paramètre 4 et vous divisez par 2 le paramètre 3 pour avoir une triple croche, par 4 pour avoir une quadruple croche, puis par 16, par 32, etc.
Par contre pour avoir des vitesses inférieures à la ronde vous multipliez le paramètre 4 par 2, et vous retranchez 1. Quelques exemples :
Pour jouer une ronde en si1, tapez : 255, 50, 91,9
Pour jouer une croche en la # 2, tapez : 0, 140, 200, 2
Pour jouer une quadruple croche en la1, tapez : 255, 100, 20, 1
Pour jouer une deux ronde liées en mi2, tapez : 0, 220, 127, 17
Etc.

Attention !
Respectez l'ordre des valeurs : paramètre 1, paramètre 2, coefficient de rattrapage, durée de la note.
Maintenant sortons de la peau du musicien, pour revêtir celle de l'informaticien ; comment utiliser tous ces paramètres ? (Attention ! dans la suite de ce paragraphe, les adresses sont données en hexadécimal ($) et les paramètres en décimal (#)). D'abord, sachez que le programme se charge de $C280 à $C2FF, alors que les paramètres sont rangés de $C404 à $C4FF ce qui fait quatre paramètres par note, 63 notes maximum.
Les notes seront donc rangées ainsi :

Première note	$C404	param 1
	$C405	param 2
	$C406	coef, rattrap
	$C407	durée note
Deuxième note	$C408	param 1
	$C409	param 2
	$C40A	coef, rattrap
	$C40B	durée note
Troisième note	$C40C	param 1
	$C40D	param 2
	$C40E	coef, rattrap
	$C40F	durée note
...		
Soixante-troisième note	$C4FC	param 1
	$C4FD	param 2
	$C4FE	coef rattrap
	$C4FF	durée note

Bien sûr, vous n'êtes pas obligé de mettre 63

notes, vous pouvez en mettre moins, mais pas plus ! Vous devez indiquer ce nombre en pokant à l'adresse : Poke$B806, # (nombre de note). N'oubliez pas de la faire sinon, vous pourriez avoir de très désagréables surprises. Ceci étant, le programme machine charge, les paramètres et le nombre de notes aussi, vous pouvez faire CALL$C292.
Si vous n'avez pas bien suivi jusqu'ici, recommencer depuis le début du chapitre, et ceci jusqu'à ce que vous compreniez, c'est très important ! Les autres suivez moi ! Après ces hors d'œuvres théoriques, nous allons nous enfoncer dans un dédale d'octets. Voici comment fonctionne ce programme.
Le P.C. 1251 est équipé d'un buzzer. Celui-ci peut prendre deux positions respectives, une position haute que l'on appellera 1 et une position basse que l'on appellera 2. Quand le buzzer passe de la position haute à la position basse et réciproquement, il émet un petit bruit. C'est une suite très rapide de ces petits bruits qui va provoquer un son. Plus la fréquence de passage de 1 à 2 et 2 à 1 sera grande et plus la note sera élevée. Pour réduire, ou augmenter cette fréquence il faut utiliser des wait'machine, qui ralentissent la fréquence d'oscillation. Ces valeurs de wait sont données par les deux premiers paramètres que nous avons étudiés tout à l'heure. Mais plus ces wait sont courts, et plus la note est jouée rapidement. En effet, pour faire une analogie avec le basic si vous faites une boucle : FORI = 1 to 100 : WAIT : NEXTI. On s'aperçoit alors que plus X sera petit et plus ce programme sera effectué rapidement. C'est la même chose en langage machine, d'où l'utilité d'une constante de rattrapage de temps pour que toutes les notes soient jouées pendant un même temps. C'est le troisième paramètre que nous avons étudié. Remarquez que plus la fréquence est élevée, donc plus la note est haute, plus les paramètres wait diminuent et plus la constante de rattrapage augmente, vérifiant ainsi ce que l'on vient de dire. Maintenant que nous avons toutes les notes de la gamme, il faut pouvoir les jouer plus ou moins longtemps les unes par rapport aux autres. C'est là qu'intervient le dernier paramètre. Dans la fin du programme il y a une boucle qui selon la valeur du dernier paramètre, joue 1 fois, 2 fois, (etc.), la note.

Encore une chose :
Pour une application personnelle de ce programme et si vous ne voulez pas combiner l'affichage qui fera partie du deuxième chapitre, placez les Pokes suivants : Poke$C2F5, $CE, $CE, $CE. J'espère que fort de tous ces renseignements, vous écrirez de jolies mélodies sur votre P.C.

PROGRAMME 2 : ROUTINE DE DEFILEMENT A L'ECRAN

Encore un programme que vous pourrez réutiliser. Attention ! Ce programme n'est pas très long, mais les paramètres prennent eux de la place. La routine commence $C000 à $C0FF, $C3000 à $C3FF et finit en $C500 à $C5CF. Le but de ce programme est de faire défiler à l'écran 5 lignes de 10 caractères, chaque caractère étant défini par 5 octets. Voici comment définir votre dessin : vous prenez une feuille quadrillée 5 x 5 (petits carreaux). Vous dessinez un grand rectangle de 50 x 40 petits carreaux. Puis vous divisez horizontalement en 10 et verticalement en 5. Vous devez alors obtenir 5 lignes horizontales constituées de 40 barreaux de 8 carreaux verticaux. Un barreau de 8 carreaux représente un octet, ainsi qu'un afficheur de votre P.C. Mais l'afficheur du 1251 ne comporte que 7 points verticaux ! Mais c'est là toute l'astuce de ce programme. Vous verrez non seulement défiler le 8e bit, mais aussi tous les octets qui se trouvent en dessous de la première ligne, sans qu'il n'y ait d'interruption.
Le codage des octets : comme vous le savez, un octet est composé de 8 bits. Seul 7 apparaissent à l'écran mais le 8e existe. C'est grâce au scrolling vertical que l'on pourra le faire apparaitre à l'écran. Comme le montre la figure 2, le bit de plus haut rang est celui du bas et celui qui a le rang le plus faible est celui du haut.

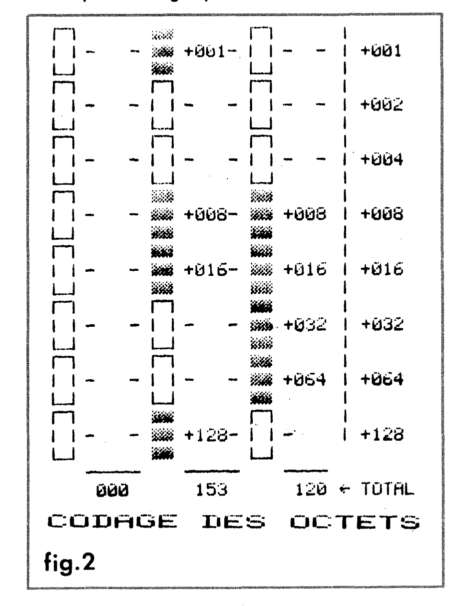


fig.2

Pour coder un octet, vous mettez la valeur de chaque bit et vous additionnez celles pour lesquelles, le bit doit apparaitre en noir à l'écran. Une fois que vous connaissez la valeur des 250 octets composant le dessin, il ne reste plus qu'à les ranger en mémoire (figure 3). Le rangement se fait à partir de l'adresse 49950 et s'effectue de haut à droite. Voilà, c'est tout simple pourvu que vous ayez quelques bases de binaire.

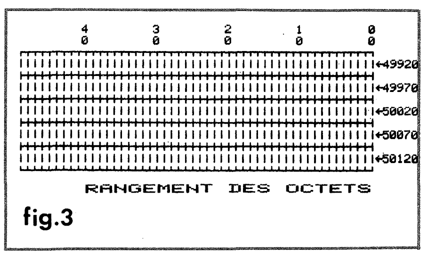


fig.3

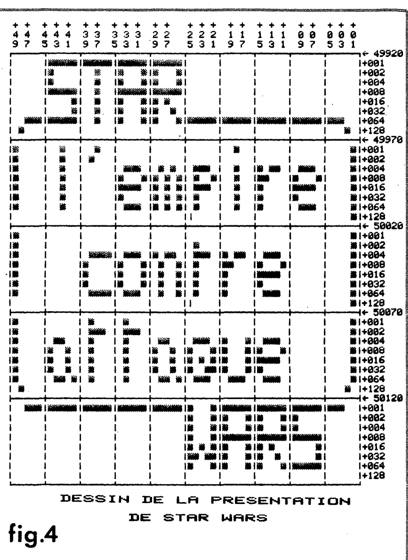


fig.4

Explication théorique du programme. Voyons tout d'abord la fonction 'SR'. Quand on applique la fonction SR a un octet, tous les bits de celui-ci sont déplacés d'un écran. C'est-à-dire que le bit rang 7, devient le bit de rang 6, celui de 6 devient 5, et., jusqu'à celui de rang 0 qui se charge dans un bit spécial appelé 'CARRY'.

C'est la méthode qui est utilisée ici. On prend les 50 premiers octets. On les décale vers le haut et pour chaque octets on regarde CARRY. Si il se charge à 0 on passe au suivant, mais si il se charge à 1, on fait un OU logique avec l'octet d'affichage correspondant, pour mettre a un bit 217. Une fois tous les octets décalés, on fait même chose avec l'afficheur, ce qui fait qu'à chaque fois que vous allez appeler la routine, vous allez voir apparaitre par le bas les octets de la première ligne. Au bout de 8 appels, la première ligne sera épuisée et on

commence la deuxième et ainsi de suite, jusqu'à la fin de la 5e.
Vous trouvez celà compliqué ?
Oh, je vous comprends très bien. Mais le principe est assez simple.

CONCLUSION

Le programme 3, ainsi que le 4, sont le jeu par lui-même. Etant très spécifique à ce jeu, il serait superflu de le commenter. Mais pour les obstinés du langage machine, je leur fournis quand même le listing en langage machine commenté.
Maintenant voyons comment enregistrer ce programme. Le grand avantage de STAR WARS est que ceux qui ne possèdent pas le CE125 peuvent aussi y jouer. Il suffit qu'ils tapent les programmes les un après les autres et qu'ils fassent 'RUN' après avoir fini chaque programme. Pour ceux qui possèdent le magnétocassette, ils doivent enregistrer chaque programme. Mais réenregistrer les 4 programmes à chaque fois, peut paraitre long à certains. Voici une astuce pour ceux dont le fait de faire travailler les autres les fatiguent plus que si c'était eux qui faisaient le travail. 'STAR WARS' occupe toute la mémoire donc il suffit de déplacer le pointeur de fin de mémoire basic à son extrême valeur. Une fois le programme entièrement tapé et fonctionnant parfaitement, tapez :
POKE&C6E3, &CF : POKE &C6E4, &C5 : POKE &C5CF, 255, CSAVE

Ainsi, votre P.C. va enregistrer 3486 octets sans distinction L.M. ou basic. Deux inconvénients mineurs. Par ce fait, l'initialisation de l'affichage n'est pas faite et il se peut que le défilement écran n'apparaisse qu'au bout de 3 run. Deuxième inconvénient, quand vous aller lister le programme en mode 'PRO', il se peut que vous ayez des signes bizares sur l'écran. Cela vient du fait que l'interpréteur basic essaye de coder les octets L.M., alors il a quelques problèmes. Mais en aucun cas, le déroulement du programme ne peut être affecté par ce mode de chargement. Voilà l'ensemble des choses que l'on peut dire sur ce programme. Encore une petite indication. Mon record au niveau 5 est de 504 survivants. Les touches de déplacement sont :
DEL et INS et celle de tir : ↑

J'espère que vous apprécierez ce jeu, et que les routines graphiques et sonores déclancheront des vocations...

```
------------- LISTING EN LANGAGE MACHINE DU PROGRAMME 1 ------------

C280 LIB   $ FF   03 FF    : CHARGE $FF DANS ´B´
C282 LIA   $ 11   02 11    : CHARGE $11 DANS ´A´
C284 CAL   $ 1FB1 FF B1    : APPELLE LA SOUS ROUTINE EN $1FB1
C286 WAIT  $ FF   4E FF    : EFFECTU $FF CYCLES MACHINE
C288 WAIT  $ FF   4E FF    : EFFECTU $FF CYCLES MACHINE
C28A LIA   $ 01   02 01    : CHARGE $01 DANS ´A´
C28C CAL   $1FB1  FF B1    : APPELLE LA SOUS ROUTINE EN $1F1B
C28E DECB         C3       : DECREMENTE ´B´
C28F JRNZM $ 0E   29 0E    : SI A<>0 ALLER EN $C280
C291 RTS          37       : RETOURNER AU DERNIER CALL EFFECTUE
C292 LIDP  $ B806 10 B8 06 : CHARGE $B806 DANS ´DP´
C295 LDD          57       : CHARGE DANS ´A´ LA VALEUR DE L´ADD ´DP´
C296 DECA         43       : DECREMENTE ´A´
C297 STD          52       : CHARGE ´A´ DANS ´DP´
C298 JRNZP $ 02   28 02    : SI Z<>0 VA EN $C29B
C29A RTS          37       : RETOURNE AU DERNIER CALL EFFECTUE
C29B LIDP  $C2AF  10 C2 AF : CHARGE $C2AF DANS ´DP´
C29E LDD          57       : CHARGE DANS ´A´ LA VALEUR DE L´ADD ´DP´
C29F ADIA$04      74 04    : ADDITIONNE A ´A´ LA VALEUR $04
C2A1 STA          52       : CHARGE ´A´ A L´ADD ´DP´
C2A2 LIDP  $ C2B7 10 C2 B7 : CHARGE $C2B7 DANS ´DP´
C2A5 INCA         42       : INCREMENTE ´A´
C2A6 STD          52       : CHARGE ´A´ A L´ADD ´DP´
C2A7 LIDP  $ C2BF 10 C2 BF : CHARGE $C2BF DANS ´DP´
C2AA INCA         42       : INCREMENTE ´A´
C2AB STD          52       : CHARGE ´A´ A L´ADD ´DP´
C2AC NOP          CE       : NE FAIT RIEN
C2AD LIDP  $ C404 10 C4 04 : CHARGE $C404 DAND ´DP´
C2B0 LDD          57       : CHARGE DANS ´A´ LA VALEUR DE L´ADD ´DP´
C2B1 LIDP  $ C287 10 C2 87 : CHARGE $C287 DANS ´DP´
C2B4 STD          52       : CHARGE ´A´ A L´ADD ´DP´
C2B5 LIDP  $ C405 10 C4 05 : CHARGE $C405 DANS ´DP´
C2B8 LDD          57       : CHARGE DANS ´A´ LA VALEUR DE L´ADD ´DP´
C2B9 LIDP  $ C289 10 C2 89 : CHARGE $C289 DAND ´A´
C2BC STD          52       : CHARGE ´A´ A L´ADD ´DP´
C2BD LIDP  $ C406 10 C4 06 : CHARGE $C406 DANS ´DP´
C2C0 LDD          57       : CHARGE DANS ´A´ LA VALEUR DE L´ADD ´DP´
C2C1 LIDP  $ C291 10 C2 81   CHARGE $C291 DANS ´DP´
C2C4 STD          52       : CHARGE ´A´ A L´ADD ´DP´
C2C5 LIDP  $ C2BF 10 C2 BF : CHARGE $C2BF DANS ´DP´
C2C8 LDD          57       : CHARGE DANS ´A´ LA VALEUR DE L´ADD ´DP´
C2C9 INCA         42       : ICREMENTE ´A´
C2CA LIDP  $ C2D0 10 C2 D0 : CHARGE $C2D0 DANS ´DP´
C2CD STD          52       : CHARGE ´A´ A L´ADD ´DP´
C2CE LIDP  $ C403 10 C4 03 : CHARGE $C403 DANS ´DP´
C2D1 LDD          57       : CHARGE DANS ´A´ LA VALEUR DE L´ADD ´DP´
C2D2 LIDP  $ B807 10 B8 07   CHARGE $B807 DANS ´DP´
C2D5 STD          52       : CHARGE ´A´ A L´ADD ´DP´
C2D6 LIDP  $ B807 10 B8 07 : CHARGE $B807 DANS ´DP´
C2D9 LDD          57       : CHARGE DANS ´A´ LA VALEUR DE L´ADD ´DP´
C2DA DECA         43       : DECREMENTE ´A´
C2DB STD          52       : CHARGE ´A´ A L´ADD ´DP´
C2DC JRNZP $ 04   28 04    : SI A<>0 ALLER EN $C2E1
C2DE JP    $ C2E7 79 C2 E7 : VA EN $C2E7
```

```
------------- LISTING EN LANGAGE MACHINE DU PROGRAMME 2 --------------

C500 LIDP  $ F845 10 F8 45 : CHARGE $F845 DANS ´DP´
C503 RC           D1       : MET ´Z´=1 , ´C´=0
C504 LDD          57       : CHARGE DANS ´A´ LA VALEUR DE L´ADD ´DP´
C505 SR           D2         EFFECTU UNE ROTATION A DROITE
C506 STD          52       : CHARGE ´A´ A L´ADD ´DP´
C507 LIDP  $ C000 10 C0 00 : CHARGE $C000 DANS ´DP´
C50A LDD          57       : CHARGE DANS ´A´ LA VALEUR DE L´ADD ´DP
C50B SR           D2         EFFECTU UNE ROTATION A DROITE
C50C STD          52       : CHARGE ´A´ A L´ADD ´DP´
C50D JRNCP $ 06   2A 06    : SI ´C´=0 ALLER EN $C514
C50F LIDP  $ F845 10 F8 45 : CHARGE $F845 DANS ´DP´
C512 ORID  $ 80   D5 80    : EFFECTU UN OU ENTRE $80 ET L´ADD ´DP´
C514 LIDP  $ C509 10 C5 09 : CHARGE $C509 DANS ´DP´
C517 LDD          57       : CHARGE DANS ´A´ LA VALEUR DE L´ADD ´DP´
C518 INCA         42       : INCREMENTE ´A´
C519 STD          52       : CHARGE ´A´ A L´ADD ´DP´
C51A LIDP  $ C502 10 C5 02 : CHARGE $C502 DANS ´DP´
C51D LDD          57       : CHARGE DANS ´A´ LA VALEUR DE L´ADD ´DP´
C51E INCA         42       : INCREMENTE ´A´
C51F STD          52       : CHARGE ´A´ A L´ADD ´DP´
C520 LIDP  $ C511 10 C5 11 : CHARGE $C511 A DANS ´DP´
C523 STD          52       : CHARGE ´A´ A L´ADD ´DP´
C524 CPIA  $ 77   67 77    : COMPARE ´A´ ET $77
C526 JPNZ  $ C500 7C C5 00 : SI A<>$77 ALLER EN $C500
C529 LIDP  $ B809 10 B8 09 : CHARGE $B809 DANS ´DP´
C52C LDD          57       : CHARGE DANS ´A´ LA VALEUR DE L´ADD ´DP´
C52D DECA         43       : DECREMETE ´A´
C52E STD          52       : CHARGE ´A´ A L´ADD ´DP´
C52F JRZP  $ C546 7E C5 46 : SI ´A´=0 ALLER EN $C546
C532 LIDP  $ C502 10 C5 02 : CAHRGE $C502 DAND ´DP´
C535 LIA   $ 45   02 45    : CHARGE $45 DANS ´A´
C537 STD          52       : CHARGE ´A´ A L´ADD ´DP´
C538 LIDP  $ C511 10 C5 11 : CHARGE $C511 DANS ´DP´
C53B STD          52       : CHARHE ´A´ A L´ADD ´DP´
C53C LIDP  $ C509 10 C5 09 : CHARGE $C509 DANS ´DP´
C53F LDD          57       : CHARGE DANS ´A´ LA VALEUR DE L´ADD ´DP´
C540 SBIA  $ 32   75 32    : SOUSTRAIT $32 A ´A´
C542 STD          52       : CHARGE ´A´ A L´ADD ´DP´
C543 RTS          37       : RETOURNE AU DERNIER CALL EFFECTUE
```

```
C544 NOP          CE        : NE FAIT RIEN
C545 NOP          CE        : NE FAIT RIEN
C546 LIDP  $ C502 10 C5 02 : CHARGE $C502 DANS 'DP'
C549 LIA   $ 45   02 45     : CHARGE $45 DANS 'DP'
C54B STD          52        : CHARGE 'A' A L'ADD 'DP'
C54C LIDP  $ C511 10 C5 11 : CHARGE $C511 DANS 'DP'
C54F STD          52        : CHARGE 'A' A L'ADD 'DP'
C550 LIDP  $ B809 10 B8 09 : CHARGE $B809 A L'ADD 'DP'
C553 LIA   $ 08   02 08     : CHARGE $08 DANS 'A'
C555 STD          52        : CHARGE 'A' A L'ADD 'DP'
C556 RTS          37        : RETOURNE AU DERNIER CALL EFFECTUE
-----------------------------------------------------------------------
C280 LIB   $ FF   03 FF     : CHARGE $FF DANS 'A'
C282 LIA   $ 11   02 11     : CHARGE $FF DANS 'B'
C284 CAL   $ 1FB1 FF B1     : APPELE LE SOUS PROGRAMME EN $1FB1
C286 WAIT  $ FF   4E FF     : EFFECTUE $FF CYCLE MACHINE
C288 WAIT  $ FF   4E FF     : EFFECTUE $FF CYCLE MACHINE
C28A LIA   $ 01   02 01     : CHARGE $01 DANS 'A'
C28C CAL   $1FB1  FF B1     : APPELE LE SOUS PROGRAMME EN $1FB1
C28E DECB         C3        : DECREMENTE 'B'
C28F JRNZM $ 0E   29 0E     : SI A<>0 ALLER EN $C280
```

```
-----------------------------------------------------------------------
------------- LISTING EN LANGAGE MACHINE DU PROGRAMME 3 ---------------
-----------------------------------------------------------------------
C100 LIDP  $ B800 10 B8 00 : CHARGE $B800 DANS 'DP'
C103 LDD          57        : CHARGE DANS 'A' LA VALEUR DE L'ADD 'DP'
C104 DECA         43        : DECREMENTE 'A'
C105 STD          52        : CHARGE 'A' DANS 'DP'
C106 JRNZP $ 0A   28 0A     : SI Z<>0 VA EN $C111
C108 LIDL  $ 01   11 01     : CHARGE $01 DANS 'DPL'
C10A LDD          57        : CHARGE LA VALEUR DE 'A' DANS 'DP'
C10B LIDL  $ 00   11 00     : CHARGE $00 DANS 'DPL'
C10D STD          52        : CHARGE 'A' A L'ADD 'DP'
C10E CALL $C240   78 C2 40 : APPELLE LE SOUS PROGRAMME EN $C240
C111 RTS          37        : RETOURNE AU DERNIER CALL EFFECTUE
C11C LIDP  $ C131 10 C1 31 : CHARGE $C131 DANS 'DP'
C11F LDD          57        : CHARGE DANS 'A' LA VALEUR DE L'ADD 'DP'
C120 INCA         42        : INCREMENTE 'A'
C121 STD          52        : CHARGE 'A' DANS 'DP'
C122 LIB   $ 00   03 00     : CHARGE DANS 'B' LA VALEUR $00
C124 LIDP  $ F801 10 F8 01 : CHARGE DANS 'DP' LA VALEUR $F801
C127 LDD          57        : CHARGE DANS 'A' LA VALEUR DE L'ADD 'DP'
C128 EXAB         DA        : ECHANGE LES VALEURS DE 'A' ET 'B'
C129 STD          57        : CHARGE 'A' A L'ADD 'DP'
C12A LIDP  $ C126 10 C1 26 : CHARGE DANS 'DP' LA VALEUR $C126
C12D LDD          57        : CHARGE DANS 'A' LA VALEUR DE L'ADD 'DP'
C12E INCA         42        : INCREMENTE 'A'
C12F STD          52        : CHARGE 'A' A L'ADD 'DP'
C130 CPIA  $ 08   67 08     : COMPARE 'A' AVEC LA VALEUR $08
C132 JRNZ  $ C124 7C C1 24 : SI Z=0 VA EN $C124
C135 LDD          57        : CHARGE DANS 'A' LA VALEUR DE L'ADD 'DP'
C136 SBIA  $ 07   7507      : SOUSTRAIT $07 A 'A'
C138 STD          57        : CHARGE 'A' A L'ADD 'DP'
C139 RTS          37        : RETOURNE AU DERNIER CALL EFFECTUE
C140 LIDP  $ C151 10 C1 51 : CHARGE LA VALEUR $C151 DANS 'DP'
C143 LDD          57        : CHARGE DANS 'A' LA VALEUR DE L'ADD 'DP'
C144 DECA         43        : DECREMENTE 'A'
C145 STD          52        : CHARGE 'A' A L'ADD 'DP'
C146 JPNZ  $, C14F 7C C1 4F : SI Z=0 VA EN $C14F
C149 LIA   $ 3B   02 3B     : CHARGE DANS 'A' LA VALEUR $3B
C14B STD          52        : CHARGE 'A' A L'ADD 'DP'
C14C RTS          37        : RETOURNE AU DERNIER CALL EFFECTUE
C14D NOP          CE        : NE FAIT RIEN
C14E NOP          CE        : NE FAIT RIEN
C14F LIDP  $ F83B 10 F8 3B : CHARGE LA VALEUR $F83B A L'ADD 'DP'
C152 LDD          57        : CHARGE DANS 'A' LA VALEUR DE L'ADD 'DP'
C153 TSIA  $ 04   66 04     : ET LOGIQUE ENTRE A ET $04
C155 JPZ   $ C140 7E C1 40 : SI Z<>0 VA EN $$C140
C158 LDD          57        : CHARGE DANS 'A' LA VALEUR DE L'ADD 'DP'
C159 SBIA  $ 04   75 04     : SOUSTRAIT $04 A 'A'
C15B STD          52        : CHARGE 'A' A L'ADD 'DP'
C15C LIDP  $ C151 10 C1 51 : CHARGE $C151 DANS 'DP'
C15F CALL  $ C165 10 C1 65 : APPELLE LE SOUS PROGRAMME EN $C165
C162 JP    $ C149 79 C1 49 : VA EN $C149
C165 LIDP  $ C171 10 C1 71 : CHARGE $C171 DANS 'DP'
C168 LIB   $ 00   03 00     : CHARGE $00 DANS 'B'
C16A LP    $ 00   83        : CHARGE $03 DANS 'P'
C16B MVDM         53        : CHARGE 'P' DANS 'DP'
C16C LIA   $ 11   02 11     : CHARGE $11 DANS 'A'
C16E CALL  $ 1FB1 FF B1     : APPELLE LE SOUS PROGRAMME $1FB1
C170 WAIT  $ 03   4E 03     : EFFECTUE 3 CYCLES MACHINES
C172 LIA   $ 01   02 01     : CHARGE $01 DANS 'A'
C174 CALL  $ 1FB1 FF B1     : APPELLE LE SOUS PROGRAMME $1FB1
C176 INCB         C2        : INCREMENTE 'B'
C177 JRNZM $ E0   29 0E     : SI Z=0 VA EN $C165
C179 RTS          37        : RETOURNE AU DERNIER CALL EFFECTUE
C185 LIDP  $ C191 10 C1 91 : CHARGE $C191 DANS 'DP'
C188 LIB   $ 00   03 00     : CHARGE $00 DANS 'B'
C18A LP    $ 00   83        : CHARGE $03 DANS 'P'
C18B MVDM         53        : CHARGE 'P' DANS 'DP'
C18C LIA   $ 11   02 11     : CHARGE $02 DANS 'A'
C18E CALL  $ 1FB1 FF B1     : APPELLE LE SOUS PROGRAMME $1FB1
C190 WAIT  $ 03   4E 03     : EFFECTUE 3 CYCLES MACHINES
C192 LIA   $ 01   02 01     : CHARGE $01 DANS 'A'
C194 CALL  $ 1FB1 FF B1     : APPELLE LE SOUS PROGRAMME $1FB1
C196 INCB         C3        : INCREMENTE.'B'
C197 JRNZM $ 0E   29 0E     : SI Z=0 VA EN $C185
C199 RTS          37        : RETOURNE AU DERNIER CALL EFFECTUE
C19A LIB   $ 30   03 30     : CHARGE $30 DANS 'B'
C19C LIA   $ 10   02 10     : CHARGE $10 DANS 'A'
C19E CALL  $ 1FB1 FF B1     : APPELLE LE SOUS PROGRAMME $1FB1
C1A0 WAIT  $ FF   4E FF     : EFFECTUE 255 CYCLES MACHINES
C1A2 LIA   $ 00   02 00     : CHARGE $00 DANS 'A'
C1A4 CALL  $ 1FB1 FF B1     : APPELLE LE SOUS PROGRAMME $1FB1
C1A6 DECB         C3        : DECREMENTE 'B'
C1A7 JRNZM $ 0C   29 0C     : SI Z=0 VA EN C19A
C1A9 RTS          37        : RETOURNE AU DERNIER CALL EFFECTUE
C1B0 CALL  $ C100 80 C1 00 : APPELLE LE SOUS PROGRAMME $C100
C1B3 LIDP  $ B802 10 B8 02 : CHARGE $B802 DANS 'DP'
C1B7 LDD          57        : CHARGE DANS 'A' LA VALEUR DE L'AD 'DP'
C1B8 DECA         43        : DECREMENTE 'A'
C1B9 STD          52        : CHARGE 'A' A L'ADD 'DP'
C1BA JPNZ  $ C1B0 7C C1 B0 : SI Z=0 VA EN $C1B0
C1BD LIA   $ 20   02 20     : CAHRGE $20 DANS 'A'
C1BF STD          52        : CHARGE 'A' A L'ADD 'DP'
C1C0.LIDP  $ B803 10 B8 03 : CHARGE $B803 DANS 'DP'
C1C3 LIP   $ 5C   12 5C     : CHARGE  $5C DANS 'P'
C1C5 LII   $ 01   00 01     : CHARGE $ 01 DANS 'I'
C1C7 MVWD         18        : CHARGE 'DP' DANS 'P'
C1C8 OUTA         DD        : CHARGE $5C DANS LE PORT IA
C1C9 OUTB         4C        : CHARGE $5D DANS LE PORT IB
C1CA DECP         51        : DECREMENTE 'P'
C1CB CMPA         C7        : COMPARE 'P' ET 'A'
C1CC CPIA  $ 01   00 01     : COMPARE 'A' ET $01
C1CE JPNZ  $ C1E6 7C C1 E6 : SI Z=0 VA EN $C1E6
C1D1 LIDP  $ F87B 10 F8 7B : CHARGE $ F87B DANS 'DP'
C1D4 LDD          57        : CHARGE DANS 'A' LA VALEUR DE L'ADD 'DP'
C1D5 LIDP  $ C154 10 C1 54 : CHARGE $C154 DAN6 'DP'
C1D8 STD          52        : CHARGE 'A' A L'ADD 'DP'
C1D9 LIDP  $ C15A 10 C1 5A : CHARGE $C15A DANS 'DP'
C1DC STD          52        : CHARGE 'A' A L'ADD 'DP'
C1DD CALL  $ C185 78 C1 85 : APPELLE LE SOUS PROGRAMME $C185
C1E0 CALL  $ C140 78 C1 40 : APPELLE LE SOUS PROGRAMME $C140
C1E3 CALL  $ 11E0 78 11 E0 : APPELLE LE SOUS PROGRAMME $11E0
C1E6 CALL  $ C100 78 C1 00 : APPELLE LE SOUS PROGRAMME $C100
C1E9 LIDP  $ B805 10 B8 05 : CHARGE $B805 DANS 'DP'
C1EC LDD          57        : CHARGE DANS 'A' LA VALEUR DE L'ADD 'DP'
C1ED DECA         43        : DECREMENTE 'A'
C1EE STD          52        : CHARGE 'A' A L'ADD 'DP'
C1EF JRNZ  $ C1B0 7C C1 B0 : SI Z=0 VA EN $C1B0
C1F2 LIA   $ 10   02 10     : CHARGE $10 DANS 'A'
C1F4 STD          52        : CHARGE 'A' A L'ADD 'DP'
C1F5 CALL  $ C11C 78 C1 1C : APPELLE LE SOUS PROGRAMME $C11C
C1F8 NOP          CE        : NE FAIT RIEN
C1F9 LIDP  $ C131 10 C1 31 : CHARGE $C131 DANS 'DP'
C1FC LDD          57        : CHARGE DANS 'A' LA VALEUR DE L'ADD 'DP'
C1FD CPIA  $ 3B   67 3B     : COMPARE 'A' AVEC $3B
C1FF JRNZ  $ C203 7C C2 03 : SI Z=0 VA EN $C203
C202 RTS          37        : RETOURNE AU DERNIER CALL EFFECTUE
C203 LIDP  $ F83B 10 F8 3B : CHARGE $F83B DAND 'DP'
C206 LDD          57        : CHARGE DANS 'A' LA VALEUR DE L'ADD 'DP'
C207 CPIA  $ 00   67 00     : COMPARE 'A' AVEC $00
C209 JRNZ  $ C218 7C C2 18 : SI Z=0 VA EN $C218
C20C LIDP  $ C205 10 C2 05 : CHARGE $C205 DANS 'DP'
C20F LDD          57        : CHARGE DANS 'A' LA VALEUR DE L'ADD 'DP'
C210 DECA         43        : DECREMENTE 'A'
C211 STA          52        : CHARGE 'A' L'ADD 'DP'
C212 CPIA  $ 00   67 00     : COMPARE 'A' ET $00
C214 JRNZ  $ C203 7C C2 03 : SI Z=0 VA EN $C203
C217 RTS          37        : RETOURNE AU DERNIER CALL EFFECTUE
C218 LIDP  $ C205 10 C2 05 : CHARGE $C205 DANS 'DP'
C21B LIA   $ 3B   02 3B     : CHARGE $3B DANS 'A'
C21D STD          52        : CHARGE 'A' A L'ADD 'DP'
C21E JP    $ C1B0 79 C1 B0 : VA EN $C1B0
C221 RTS          37        : RETOURNE AU DERNIER CALL EFFECTUE
C240 LIDP  $ B803 10 B8 03 : CHARGE $B803 DAND 'DP'
C243 LIP   $ 5C   12 5C     : CHARGE $5C DANS 'P'
C245 LII   $ 01   00 01     : CHARGE $01 DANS 'I'
C247 MVWD         18        : CHARGE 'DP' DANS 'P'
C248 OUTA         5D        : CHARGE $5C DANS LE PORT IA
C249 OUTB         DD        : CHARGE $5D DANS LE PORT IB
C24A INA          4C        : CHARGE LA VALEUR DU PORT IA DANS 'A'
C24B DECP         51        : DECREMENTE 'P'
C24C CPMA         C7        : COMPARE 'P' ET 'A'
C24D CPIA  $ 02   67 02     : COMPARE 'A' ET $02
C24F JRNZ  $ C260 7C C2 60 : SI Z=0 VA EN $C260
C252 LIDP  $ F87B 10 F8 7B : CHARGE $F87B DANS 'DP'
C255 LDD          57        : CHARGE DANS 'A' LA VALEUR DE L'ADD 'DP'
C256 SL           5A        : ROTATION LOGIQUE A GAUCHE
C257 STD          52        : CHARGE 'A' A L'ADD 'DP'
C258 CPIA  $ 80   67 80     : COMPARE 'A' AVEC $80
C25A JRNZP $ 04   28 04     : SI Z=0 VA EN C25F
C25C LIA   $ 01   02 01     : CHARGE $01 DANS 'A'
C25E STD          52        : CHARGE 'A' A L'ADD 'DP'
C25F RTS          37        : RETOURNE AU DERNIER CALL EFFECTUE
C260 CPIA  $ 04   67 04     : COMPARE 'A' AVEC $04
C662 JPNZ  $ C25F 7C C2 5F : SI Z=0 VA EN $C25F
C665 LIDP  $ F87B 10 F8 7B : CHARGE $F87B DANS 'DP'
C668 LDD          57        : CHARGE DANS 'A' LA VALEUR DE L'ADD 'DP'
C669 CPIA  $ 01   67 01     : COMPARE $01 ET 'A'
C66B JRNZ  $ C272 7C C2 72 : SI Z=0 VA EN $C272
C66E LIA   $ 40   02 40     : CHARGE $40 DANS 'A'
C670 STD          52        : CHARGE 'A' A L'ADD 'DP'
C671 RTS          37        : RETOURNE AU DERNIER CALL EFFECTUE
C672 SR           D2        : DECALAGE LOGIQUE A DROITE
C273 STD          57        : CHARGE 'A' A L'ADD 'DP'
C274 RTS          37        : RETOURNE AU DERNIER CALL EFFECTUE

READY.

C2E1 CALL  $ C280 78 C2 80 : APELLE LE SOUS PROGRAMME EN $C280
C2E4 JP    $ C2D6 79 C2 D6 : VA EN $C2D6
C2E7 LIDP  $ B80A 10 B8 0A : CHARGE $B806 DANS 'DP'
C2EA LDD          57        : CHARGE DANS 'A' LA VALEUR DE L'ADD 'DP'
C2EB DECA         43        : DECREMENTE 'A'
C2EC STD          52        : CHARGE 'A' A L'ADD 'DP'
C2ED JRZP  $ 04   38 04     : SI 'A'=0 ALLER EN $C2F2
C2EF JP    $ C292 79 C2 92 : ALLER EN $C292
C2F2 LIA   $ 02   02 02     : CHARGE $02 DANS 'A'
C2F4 STD          52        : CHARGE 'A' A L'ADD 'DP'
C2F5 CALL  $ C500 78 C5 00 : APELLE LA SOUS PROGRAMME EN $C500
C2F8 JP    $ C292 79 C2 92 : VA EN $C292
-----------------------------------------------------------------------
'A' = REGISTRE A (8 BITS) : 'B' = REGITRE 'B' (8 BITS)
'DP' = REGISTRE DP (16 BITS)
* SI VOUS VOULEZ UTILISEZ CE PROGRAMME TOUT SEUL SANS LA ROUTINE 2
DE DEPLACEMENT D'ECRAN , REMPLACEZ L'OCTET $C2E0 VALANT $E7 PAR $92
-----------------------------------------------------------------------
```

```
1:REM +---------------+
2:REM + PROGRAMME 1   +
3:REM +  MUSIQUE DE   +
4:REM + J.M. PRIOUX   +
5:REM + PROGRAMME DE +
6:REM +PASCAL PRIOUX +
7:REM +---------------+
10:POKE &C280,&03,&FF,&
   02,&11,&FF,&B1,&4E,&
   FF,&4E,&FF,&02,&01,&
   FF,&B1,&C3,&29
15:POKE &C290,&05,&37
100:POKE &C292,&10,&B8,&
    06,&57,&43,&52,&28,&
    02,&37,&10,&C2,&AF,&
    57,&74,&04,&52
110:POKE &C2A2,&10,&C2,&
    B7,&42,&52,&10,&C2,&
    BF,&42,&52,&CE,&10,&
    C4,&00,&57,&10
120:POKE &C2B2,&C2,&87,&
    52,&10,&C4,&01,&57,&
    10,&C2,&89,&52,&10,&
    C4,&02,&57,&10
130:POKE &C2C2,&C2,&81,&
    52,&10,&C2,&BF,&57,&
    42,&10,&C2,&D0,&52,&
    10,&C4,&03,&57
140:POKE &C2D2,&10,&B8,&
    07,&52,&10,&B8,&07,&
    57,&43,&52,&28,&04,&
    79,&C2,&E7,&78
150:POKE &C2E2,&C2,&80,&
    79,&C2,&D6,&10,&B8,&
    0A,&57,&43,&52,&38,&
    04,&79,&C2,&92
160:POKE &C2F2,&02,&02,&
    52,&78,&C5,&00,&79,&
    C2,&92
200:POKE &C404,255,20,10
    2,5
202:POKE &C408,0,170,165
    ,9
204:POKE &C40C,0,200,140
    ,2
206:POKE &C410,0,220,127
    ,3
208:POKE &C414,0,250,122
    ,2
210:POKE &C418,0,110,255
    ,5
212:POKE &C41C,0,170,165
    ,9
214:POKE &C420,0,200,140
    ,2
216:POKE &C424,0,220,127
    ,3
218:POKE &C428,0,250,112
    ,2
220:POKE &C42C,0,110,255
    ,5
222:POKE &C430,0,170,165
    ,9
224:POKE &C434,0,200,140
    ,2
226:POKE &C438,0,220,127
    ,3
228:POKE &C43C,0,200,140
    ,2
230:POKE &C440,0,250,112
    ,9
232:POKE &C444,255,150,6
    9,3
234:POKE &C448,255,100,7
    9,5
236:POKE &C44C,0,200,140
    ,2
238:POKE &C450,0,220,127
    ,2
240:POKE &C454,0,255,112
    ,2
242:POKE &C458,255,20,10
    2,3
244:POKE &C45C,250,0,56,
    2
246:POKE &C460,220,0,63,
    2
248:POKE &C464,250,0,112
    ,2
250:POKE &C468,255,100,7
    9,2
252:POKE &C46C,255,50,91
    ,3
254:POKE &C470,255,150,6
    9,3
256:POKE &C474,255,100,7
    9,5
258:POKE &C478,0,200,140
    ,2
260:POKE &C47C,0,220,127
    ,2
262:POKE &C480,0,250,102
    ,2
264:POKE &C484,20,255,10
    2,2
266:POKE &C488,0,170,165
    ,3
268:POKE &C48C,0,250,112
    ,5
270:POKE &C490,255,150,6
    9,3
272:POKE &C494,255,100,7
    9,5
274:POKE &C498,0,200,140
    ,2
276:POKE &C49C,0,220,127
    ,2
278:POKE &C4A0,0,250,112
    ,2
280:POKE &C4A4,20,255,10
    2,3
282:POKE &C4A8,0,250,56,
    2
284:POKE &C4AC,0,220,64,
    2
286:POKE &C4B0,0,250,112
    ,2
287:POKE &C4B4,100,255,7
    9,2
288:POKE &C4B8,50,255,91
    ,5
290:POKE &C4BC,0,110,255
    ,2
292:POKE &C4C0,0,140,200
    ,2
294:POKE &C4C4,0,160,175
    ,2
296:POKE &C4C8,0,170,165
    ,2
298:POKE &C4CC,0,200,140
    ,2
300:POKE &C4D0,0,235,119
    ,2
302:POKE &C4D4,0,250,112
    ,2
304:POKE &C4D8,0,200,140
    ,2
306:POKE &C4DC,0,170,165
    ,9
910:POKE &B806,56
999:CALL &C292
```

```
10:REM +--------------+
20:REM + PROGRAMME 2 +
30:REM + ECRIT PAR    +
40:REM +PASCAL PRIOUX+
50:REM +--------------+
100:POKE &C300,0,128,64,
    64,64,64,64,64,64,64
    ,64,64,64,64,64,64
110:POKE &C310,64,64,64,
    64,64,64,64,64,64,79
    ,41,25,9,127,127,9
120:POKE &C320,9,9,127,1
    ,1,127,1,1,121,73,73
    ,73,79,64,64,64
130:POKE &C330,128,0,255
    ,0,0,0,0,8,84,84,84,
    56,0,4,4,8
140:POKE &C340,124,0,0,1
    25,0,0,8,20,20,124,0
    ,120,4,120,4,120
150:POKE &C350,8,84,84,8
    4,56,0,0,1,2,0,0,0,1
    27,0,0,0
```

```
160:POKE &C360,0,0,0,255
    ,255,0,0,0,0,0,0,0,0
    ,0,8,84
170:POKE &C370,84,84,56,
    0,4,4,8,124,0,2,2,12
    7,0,120,4,4
180:POKE &C380,120,4,56,
    68,68,68,56,68,68,68
    ,68,56,0,0,0,0
190:POKE &C390,0,0,0,0,0
    ,255,127,128,0,0,0,0
    ,0,0,0,0
200:POKE &C3A0,8,84,84,8
    4,56,60,64,64,64,60,
    56,100,84,68,56,64
210:POKE &C3B0,56,68,68,
    56,0,2,2,127,0,0,2,2
    ,127,0,64,56
220:POKE &C3C0,68,68,56,
    0,0,0,128,127,0,0,1,
    1,1,121,73,73
230:POKE &C3D0,73,79,79,
    41,25,9,127,127,9,9,
    9,127,127,32,16,32
240:POKE &C3E0,127,1,1,1
    ,1,1,1,1,1,1,1,1,1,1
    ,1,1

250:POKE &C3F0,1,1,1,1,1
    ,1,1,1,0,0,0,0,0,0,0
    ,0,0,0
300:POKE &C500,&10,&F8,&
    45,&D1,&57,&D2,&52,&
    10,&C0,&00,&57,&D2,&
    52,&2A,&06,&10
310:POKE &C510,&F8,&45,&
    D5,&80,&10,&C5,&09,&
    57,&42,&52,&10,&C5,&
    02,&57,&42,&52
320:POKE &C520,&10,&C5,&
    11,&52,&67,&77,&7C,&
    C5,&00,&10,&B8,&09,&
    57,&43,&52,&7E

330:POKE &C530,&C5,&46,&
    10,&C5,&02,&02,&45,&
    52,&10,&C5,&11,&52,&
    10,&C5,&09,&57
340:POKE &C540,&75,&32,&
    52,&37,&C5,&00,&10,&
    C5,&02,&02,&45,&52,&
    10,&C5,&11,&52
350:POKE &C550,&10,&B8,&
    09,&02,&08,&52,&37,&
    10,&B8,&09,&02,&08,&
    52,&10,&C5,&02
360:POKE &C560,&02,&45,&
    52,&10,&C5,&11,&52,&
    10,&C5,&09,&02,&00,&
    52,&10,&C3,&00
370:POKE &C570,&57,&10,&
    C0,&00,&52,&10,&C5,&
    6F,&57,&42,&52,&10,&
    C5,&73,&52,&7C
380:POKE &C580,&C5,&6D,&
    10,&B8,&06,&02,&11,&
    52,&10,&C2,&AF,&02,&
    00,&52,&78,&C2
390:POKE &C590,&92,&10,&
    B8,&06,&02,&1F,&52,&
    10,&C2,&AF,&02,&40,&
    52,&78,&C2,&92
400:POKE &C5A0,&10,&B8,&
    06,&02,&1F,&52,&10,&
    C2,&AF,&02,&40,&52,&
    78,&C2,&92,&10
410:POKE &C5B0,&B8,&06,&
    02,&0A,&52,&10,&C2,&
    AF,&02,&B8,&52,&78,&
    C2,&92,&37,&B8
420:POKE &C5C0,&06,&02,&
    0A,&52,&10,&C2,&AF,&
    02,&B8,&52,&78,&C2,&
    92,&37,&10,&B8
900:WAIT 0: PRINT "P.P.S
    OFT :": CALL &11E0
910:FOR I=1 TO 5: CALL &
    C557: NEXT I
```

```
 10:REM +-------------+
 20:REM + PROGRAMME 3 +
 30:REM +  ECRIT PAR  +
 40:REM +PASCAL PRIOUX+
 50:REM +-------------+
100:POKE &C100,&10,&B8,&
    00,&57,&43,&52,&28,&
    0A,&11,&01,&57,&11,&
    00,&52,&78,&C2
105:POKE &C110,&40,&37
110:POKE &C11C,&10,&C1,&
    31,&57,&42,&52,&03,&
    00,&10,&F8,&01,&57,&
    DA,&52,&10,&C1
115:POKE &C12C,&26,&57,&
    42,&52,&67,&08,&7C,&
    C1,&24,&57,&75,&07,&
    52,&37
120:POKE &C140,&10,&C1,&
    51,&57,&43,&52,&7C,&
    C1,&4F,&02,&3B,&52,&
    37,&CE,&CE,&10
125:POKE &C150,&F8,&3B,&
    57,&66,&04,&7E,&C1,&
    40,&57,&75,&04,&52,&
    10,&C1,&51,&78
130:POKE &C160,&C1,&65,&
    79,&C1,&49
135:POKE &C165,&10,&C1,&
    71,&03,&00,&83,&53,&
    02,&11,&FF,&B1,&4E,&
    03,&02,&01,&FF
140:POKE &C175,&B1,&C2,&
    29,&0E,&37
145:POKE &C185,&10,&C1,&
    91,&03,&80,&83,&53,&
    02,&11,&FF,&B1,&4E,&
    03,&02,&01,&FF
150:POKE &C195,&B1,&C3,&
    29,&0E,&37
155:POKE &C19A,&03,&30,&
    02,&10,&FF,&B1,&4E,&
    FF,&02,&00,&FF,&B1,&
    C3,&29,&0C,&37
160:POKE &C1B0,&78,&C1,&
    00,&10,&B8,&02,&57,&
    43,&52,&7C,&C1,&B0,&
    02,&20,&52,&10

200:POKE &C1C0,&B8,&03,&
    12,&5C,&00,&01,&18,&
    5D,&DD,&4C,&51,&C7,&
    67,&01,&7C,&C1
210:POKE &C1D0,&E6,&10,&
    F8,&7B,&57,&10,&C1,&
    54,&52,&10,&C1,&5A,&
    52,&78,&C1,&85
220:POKE &C1E0,&78,&C1,&
    40,&78,&11,&E0,&78,&
    C1,&00,&10,&B8,&05,&
    57,&43,&52,&7C

230:POKE &C1F0,&C1,&B0,&
    02,&10,&52,&78,&C1,&
    1C,&CE,&10,&C1,&31,&
    57,&67,&3B,&7C
240:POKE &C200,&C2,&03,&
    37,&10,&F8,&3B,&57,&
    67,&00,&7C,&C2,&18,&
    10,&C2,&05,&57
250:POKE &C210,&43,&52,&
    67,&00,&7C,&C2,&03,&
    37,&10,&C2,&05,&02,&
    3B,&52,&79,&C1

260:POKE &C220,&B0,&37
270:POKE &C240,&10,&B8,&
    03,&12,&5C,&00,&01,&
    18,&5D,&DD,&4C,&51,&
    C7,&67,&02,&7C
280:POKE &C250,&C2,&60,&
    10,&F8,&7B,&57,&5A,&
    52,&67,&80,&28,&04,&
    02,&01,&52,&37
290:POKE &C260,&67,&04,&
    7C,&C2,&5F,&10,&F8,&
    7B,&57,&67,&01,&7C,&
    C2,&72,&02,&40
300:POKE &C270,&52,&37,&
    D2,&52,&37

900:REM
920:POKE &B800,200,200:
    POKE &F87A,255,1:
    POKE &B803,&10,0:
    POKE &C126,01: POKE
    &C131,08
930:POKE &C205,3B: POKE
    &C1BD,100
```

```
940:WAIT 0: PRINT "
            STAR WARS":
    CALL &11E0
945:POKE &F801,24,120,24
    ,24,120,31,2
947:POKE &F87A,255,1
950:CALL &C1B0
```

```
  1:REM +--------------+
  2:REM + PROGRAMME 4  +
  3:REM +  STAR  WARS  +
  4:REM + ECRIT  PAR   +
  5:REM + PASCAL PRIOUX+
  6:REM +--------------+
  7:C=100:B=10:A=0:P=0:
    POKE &C2F5,&78,&C5,&
    00
  8:INPUT "NOUVELLE PART
    IE ? ";X$: IF LEFT$
    (X$,1)="N" LET A=0:
    GOTO 200
  9:CLEAR :C=100:B=10:
    INPUT "NIVEAU ? ( 1
    A 10 )";Q: IF Q<1 OR
    Q>10 THEN 5
 10:WAIT 0: PRINT "P.P.S
    OFT :": CALL &C557:
    FOR I=1 TO 4: FOR J=
    1 TO 10: NEXT J:
    CALL &C500: NEXT I:
    CALL &11E5
200:REM ---------------
210:POKE &B800,200,200:
    POKE &F87A,255,1:
    POKE &B803,&10,0:
    POKE &C126,1: POKE &
    C131,8
220:POKE &C205,&3B: POKE
    &C1BD,C: POKE &C1F3,
    B: GOSUB 810
230:A=A+1: WAIT 0: PRINT
    "               ATTAQU
    E ";A: CALL &11E0
240:POKE &F801,24,120,24
    ,24,120,31,2: POKE &
    F87A,255,1
250:CALL &C1B0: CALL &11
    E5
260:IF PEEK &C126>50
    THEN 300
270:C=C-Q:B= INT (C/10):
    P=P+50- PEEK &C126
280:IF C<0 THEN 400
290:GOTO 200
300:WAIT 0: PRINT "  C E
    ST FINI !":N=10:D=&B
    8: GOSUB 800
310:WAIT 100: PRINT "LA
    BASE DE L ALLIANCE":
    PRINT "EST DETRUITE
    PAR LES"
320:PRINT "FORCES DE L E
    MPIRE": PRINT "APRES
     ";A;" ASSAUTS"
330:PRINT "IL Y A ";P;"
    SURVIVANTS": GOTO 50
    0
400:WAIT 0: PRINT "L ALL
    IANCE EST SAUVEE!!!"
410:N=56:D=0: GOSUB 800
420:WAIT 99: PRINT "VOUS
     AVEZ REPOUSSE":
    PRINT A;" ASSAUTS":
    PRINT "AVEC ";P;" SU
    RVIVANTS"
430:WAIT 0: PRINT "  ENC
    ORE BRAVO !!!...":N=
    10:D=&B8: GOSUB 800
500:REM ---- FIN ----
510:IF P>R LET R=P: BEEP
    2: PRINT " !** NOUVE
    AU RECORD **!":
    INPUT "VOS INITIALES
     ? ";Z$
520:PRINT "LE RECORD EST
     DETENU PAR": PRINT
    Z$;" : ";R;"SURVIVAN
    TS"
530:INPUT "ENCORE UNE PA
    RTIE ? ";X$: IF
    LEFT$ (X$,1)="O" LET
    A=0:C=100:B=10:P=0:
    GOTO 200
540:END
800:POKE &C2F5,&CE,&CE,&
    CE: POKE &B806,N:
    POKE &C2AF,D: CALL &
    C292: POKE &C2F5,&78
    ,&C5,&00: RETURN
810:WAIT 0:N=7:D=0: IF A
    =0 THEN PRINT "   L
    EMPIRE ATTAQUE !":
    GOSUB 800: RETURN
820:PRINT "   MISSION ";
    A;" REUSSIE": GOSUB
    800: RETURN
```

FONCTION AUTO SUR PC 1251

DANS LE BULLETIN N° 10, on trouve une fonction RENUM, puis dans le N° 13 nous disposons d'un FOUILLEUR au sein du MONITEUR LM. Cette fois-ci je vous propose une AUTOnumérotation en 4 lignes BASIC. Cette fonction affiche les numéros de lignes lors de la saisie d'un programme en incrémentant. Il faut en premier lieu choisir le premier n° de ligne et la valeur de l'incrément.

MODE D'EMPLOI :

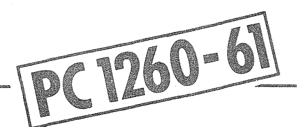
DEF A : Choix ligne et incrément.
DEF S : Affichage n° de ligne suivante etc. jusqu'à
DEF D : Fin et retour.

Inconvénient : Il faut toujours taper DEF S après ENTER. Qui dit mieux ? En LM ?

Remarque : Si une ligne existe déjà, ENTER ne l'efface pas.

V. Schadler.

```
  1:INPUT "000";B
997:"D" POKE 63550,2:
    END
998:"A" POKE 63550,2:
    INPUT "LIGNE,INCREME
    NT:";L,I:L=L-I:D=&B8
    35:E=&F83E
999:"S"L=L+I:A= INT (L/1
    00):B= INT ((L-100A)
    /10):C=L-100A-10B:
    POKE D,A+64,B+64,C+6
    4: POKE E,3: GOTO 1
```



LABELS

Si vous utilisez beaucoup d'étiquettes alphanumériques dans vos programmes, cet utilitaire est pour vous.

Il liste en effet tous les labels d'un programme ainsi que les numéros de lignes où ils se trouvent. Jouez sur le WAIT en ligne 500 pour le temps d'affichage ou utilisez LPRINT (si vous êtes l'heureux possesseur d'une imprimante. RUN 500 (ou GOTO 500) exécute le programme.

Ekkehard Otto

```
500:A= PEEK &66E1+256*
    PEEK &66E2+1: WAIT 5
510:L= PEEK (A+1)+256*
    PEEK A: IF L>65000
    END
520:IF PEEK (A+3)<>34
    GOTO 540
530:M$= CHR$ 34: FOR I=4
    TO 11:X= PEEK (A+I):
    IF X=34 LET I=12
534:M$=M$+ CHR$ X: NEXT
    I
535:PRINT L;":";M$
540:A=A+ PEEK (A+2)+3:
    GOTO 510
```

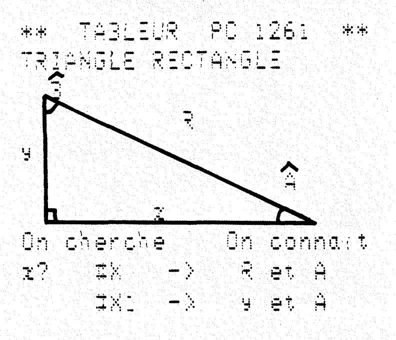
TABLEUR SUR PC-1260-1261

Une des caractéristiques les plus marquantes sur les PC-1260/1261 est la possibilité d'exploiter des équations sous tableur.
En voici quelques uns qui seront utiles je l'espère à d'autres SHARPENTIERS. Tous les angles sont en degrés minutes secondes.



Ce premier essai peut constituer une rubrique régulière, si vous le souhaitez, alors à vos PC et envoyez nous vos propres tableurs.

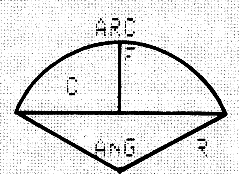
```
**  TABLEUR  PC 1261   **
TRIANGLE QUELCONQUE
      B
  y             x
C                       A
            R
On cherche      On connait
x?    ‡XQ   ->    R,C,B
      ‡XQ1  ->    y,C,A
      ‡XQ2  ->    R,y,C
y?    ‡YQ   ->    R,A,B
      ‡YQ1  ->    x,R,A
R?    ‡RQ   ->    x,y,B
      ‡RQ1  ->    x,B,C
A?    ‡AQ   ->    x,y,R
      ‡AQ1  ->    x,y,C
Exemple
R              42.00000000
Cq             46.30000000
Bq            101.00000000
XQ             31.03594124
Aq             32.30000000
y              22.98895541
**** LISTE TABLEURS ****
‡XQ=R* SIN DEG Cq/ SIN
 DEG Bq:Aq= DMS (180-(
 DEG Cq+ DEG Bq)):y=R*
 SIN DEG Aq/ SIN DEG Bq
‡XQ1=y* SIN DEG Cq/ SIN
 DEG Aq
‡XQ2=√((R^2+y^2)-(2*R*y*
 COS DEG Cq))
‡YQ=R* SIN DEG Aq/ SIN
 DEG Bq:Cq= DMS (180-(
 DEG Aq+ DEG Bq)):x=R*
 SIN DEG Cq/ SIN DEG Bq
‡YQ1=√((x^2+R^2)-(2*x*R*
 COS DEG Aq))
‡RQ=√((x^2+y^2)-(2*x*y*
 COS DEG B)):A= DMS ACS
 ((x^2+RQ^2-y^2)/(2*x*RQ
 )):C= DMS (180-( DEG A+
 DEG B))
‡RQ1=x* SIN DEG Bq/ SIN
 DEG Cq
‡AQ= DMS ACS ((x^2+R^2-y
 ^2)/(2*x*R)):B= DMS ACS
 ((y^2+x^2-R^2)/(2*x*y))
 :Cq= DMS (180-( DEG AQ+
 DEG B))
‡AQ1= DMS ASN ((y* SIN
 DEG Cq)/x)
```




```
On cherche      On connait
x?    ‡X    ->    R et A
      ‡X1   ->    y et A
```

```
y?   #Y    ->    x et A
     #Y1   ->    x et R
A?   #A    ->    x et y
     #A1   ->    y et R
Exemple:
R                42.00000000
AngA             28.30000000
X                35.15268451
y                19.08635041
AngB             61.30000000
Surface         335.46822720
**** LISTE TABLEURS ****
#X=R* COS DEG AngA:y=R*
 SIN DEG AngA:AngB= DMS
 (90- DEG AngA):Surface=
 X*y/2
#X1=y*1/ TAN DEG AngA:R=
 y/ SIN DEG AngA:AngB=
 DMS (90- DEG AngA)
#Y=x* TAN DEG AngA:R=x/
 COS DEG AngA:AngB= DMS
 (90- DEG AngA):Surface=
 x*Y/2
#Y1=√(R^2-x^2):AngA= DMS
 ACS (x/R):AngB= DMS (90
 - DEG AngA)
#A= DMS ATN (y/x):R=√(x^
 2+y^2):AngB= DMS (90-
 DEG A):Surface=x*y/2
#A1= DMS ASN (y/R):x=√(R
 ^2-y^2):AngB= DMS (90-
 DEG A1)
------------------------
```

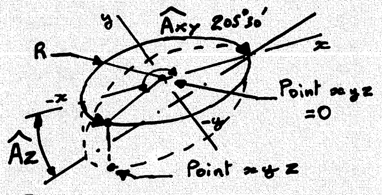
```
**  TABLEUR  PC 1261  **
CALCUL: Rayon,Corde,Arc,
Fleche,Angle(D.Min.Sec.)
NB Decimales sous forme:
  1:"D": PRINT USING "##
    #.###";"": CLS : END
------------------------
```




```
On cherche    On Connait
R?   #R   ->   C,F
     #R1  ->   C,ANG
     #R2  ->   ARC,ANG
C?   #C   ->   F,R
F?   #F   ->   C,R
     #F1  ->   R,ANG
ARC? #ARC ->   R,ANG
ANG? #ANG ->   R,ARC
```

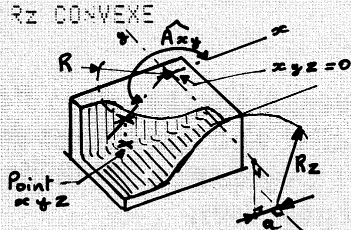
```
Exemple:
Corde            30.30000000
Fleche           10.00000000
R                16.25000000
Ang             134.45369720
Arc              38.22016920
**** LISTE TABLEURS ****
LLIST #R
#R=Corde^2/(Fleche*8)+(F
 leche/2):Ang= DMS  ASN
 (Corde/2/R)*2):Arc=(π*R
 * DEG Ang)/180
#R1=(c/2)/ SIN DEG (Ang/
 2):f=R1-√(R1^2-c^2/4):A
 rc=(π*R1* DEG Ang)/180
#R2=Arc/ DEG Ang*180/π:c
 =R2* SIN DEG (Ang/2)*2:
 f=R2-R2* COS DEG (Ang/2
 )
#C=√(2*Rayon*Fleche-Flec
 he^2)*2:Ang= DMS ( ASN
 (C/2/Rayon)*2):Arc=(π*R
 ayon* DEG Ang)/180
#F=Ray-√(Ray^2-Corde^2/4
 ):Ang= DMS ( ASN (Corde
 /2/Ray)*2):Arc=(π*Ray*
 DEG Ang)/180
#F1=R-R* COS DEG (Ang/2)
 :c=R* SIN DEG (Ang/2)*2
 :Arc=(π*R* DEG Ang)/180
#ARC=(π*R* DEG Ang)/180:
 c=2*(R* SIN ( DEG Ang/2
 )):f=R-(R* COS ( DEG An
 g/2))
#ANG= DMS (Arc/(R*π/180)
 ):c=2*(R* SIN ( DEG ANG
 /2)):f=R-(R* COS ( DEG
 ANG/2))
```

```
**  TABLEUR  PC 1261  **
CALCUL POINT x,y,z
Pente en z
On connait R,Axy,Az
```



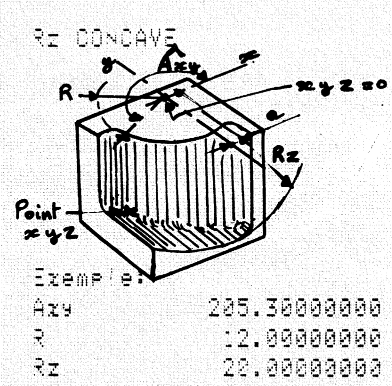

```
Exemple:
Axy             205.30000000
R                12.00000000
Az               12.15000000
ZP                0.00000000
x               -10.83102341
y                -5.16613316
z                -3.72713286
#ZP=.:x= COS DEG Axy*R:y
 = SIN DEG Axy*R:z=-(R-y
 )* TAN DEG Az
------------------------
```

```
CALCUL POINT x,y,z
Rayon en z ->   Rz>(R+a)
a=Decalage axe R et Rz
Rz CONVEXE
```




```
Exemple:
Axy             205.30000000
R                12.00000000
Rz               20.00000000
Decal             2.00000000
ZC                0.00000000
x               -10.83102341
y                -5.16613316
z                -4.65839518
#ZC=.:x= COS DEG Axy*R:y
 = SIN DEG Axy*R:z=-(Rz-
 √(Rz^2-((Decal-x)^2)))
```

```
Rz CONCAVE
```




```
Exemple:
Axy             205.30000000
R                12.00000000
Rz               22.00000000
Decal             2.00000000
ZC1               0.00000000
x               -10.83102341
y                -5.16613316
z               -15.34160481
#ZC1=.:x= COS DEG Axy*R:
 y= SIN DEG Axy*R:z=-√(R
 z^2-((Decal-x)^2))
```

J. Picot



RALLY POUR PC 1350

Si vous avez aimé OLPHOUS, voici dans le même style en plus complexe un autre jeu extrait d'une publication japonaise. Vous devez piloter une formule 1 à travers un circuit tortueux et difficile, en évitant tous les obstacles tout en jouant sur les commandes de direction et de vitesse.

Pour le rentrer dans un PC 1350, il faut disposer d'une carte mémoire d'au moins 8 Ko (le gm fait presque 5 ko) et de l'utilitaire donné par OLPHOUS dans le N° 13 page 28, à moins que vous ne préféreriez effectuer des POKEs.
Une fois entré et vérifié sauvez le sur K7 par :
CSAVE M « RALLY » : &5400,&675F
Vous pourrez dès lors le charger par : CLOAD M « RALLY »
Le jeu débute en tapant CALL &6218 ENTER.
Vous aurez droit à la présentation musicale et graphique. Par la suite CALL &621A évite la petite démo. Pour démarrer appuyez sur plusieurs touches simultanément.

8 et 2 permettent de bouger de haut en bas (en fait de gauche à droite), tandis que SPC augmente la vitesse alors que ENTER sert de pédale de frein.
Les autres commandes sont ; MODE pour figer le jeu, histoire de souffler un peu, SHIFT pour reprendre ensuite, CLS pour revenir sous Basic.
De plus ENTER permet de recommencer à nouveau le circuit.
ATTENTION aux carambolages vous n'avez que 3 voitures.
ET gare à l'alcootest !

J.F.V.

```
RALLY
(c) I/O  Pio
5400 00180055003C00AA:53
5408 003C0055004200AA:7D
5410 0042005500810AA:C2
5418 0081005500000AA:80
5420 00840055 00FF00AA:82
5428 0084005500000AA:83
5430 0026005500FB00AA:20
5438 0012005500200AA:31
5440 0000005500060AA:05
5448 0022005500230AA:44
5450 0022005500FB00AA:1C
5458 002A0055002B00AA:54
5460 0022005500060AA:27
5468 0000005500810AA:80
5470 0081005500810AA:01
5478 00FF005500810AA:7F
5480 0081005500810AA:01
5488 0000005500810AA:80
5490 0081005500420AA:C2
5498 00420055003C00AA:7D
54A0 003C005500180AA:53
54A8 0000005500000AA:FF
54B0 0000005500000AA:FF
54B8 0000005500000AA:FF
54C0 0000005500000AA:FF
54C8 180018553C003CAA:A7
54D0 3C003C5542004 2AA:FB
54D8 42004255810081AA:85
54E0 81008155000000AA:01
54E8 7F007855810084AA:FB
54F0 81008455810084AA:09
54F8 7F007855000000AA:F6
5500 030000553C00F4AA:32
5508 C0000055000000AA:BF
5510 81000E55810011AA:20
5518 FF001155810011AA:A1
5520 8100FE55000000AA:7E
5528 81008155610081AA:03
5530 42004255420042AA:07
5538 3C003C553C003CAA:EF
5540 1800185500000000:85
5548 0000000000000000:00
5550 0000000000000000:00
5558 0000000000000000:00
5560 55555500AAAAAA00:FD
5568 55555500AAAAAA00:FD
5570 55555500AAAAAA00:FD
5578 55555500AAAAAA00:FD
5580 55555500AAAAAA00:FD
5588 55555500AAAAAA00:FD
5590 55550000AAAA0000:FE
5598 55550000AAAA0000:FE
55A0 55550000AAAA0000:FE
55A8 55550007AAAA0007:0C
55B0 55550007AAAA0007:0C
55B8 55550007AAAA0007:0C
55C0 55000007AA000007:0D
55C8 55000007AA000007:0D
55D0 5500FE07AA00FE07:09
55D8 5500FE7FAA00FE7F:F9
55E0 5500FE7FAA00FE7F:F9
55E8 5500FE7FAA00FE7F:F9
55F0 5500FE7FAA000000:7C
55F8 55000000AA000000:FF
5600 55000000AA000000:FF
5608 55000000AAFE7F00:7C
5610 55000000AA000000:FF
5618 55000000AA000000:FF
5620 55000000AA000000:FF
5628 5500FE7FAA000000:7C
5630 55000000AA000000:FF
5638 55000000AA000000:FF
5640 55000000AAFE7F00:7C
5648 55000000AA000000:FF
5650 55000000AA000000:FF
5658 55000000AA00FE7F:7C
5660 55000000AA000000:FF
5668 55000000AA000000:FF
5670 55000000AAFE00FE:FB
5678 55000000AA000000:FF
5680 55000000AA000000:FF
5688 5500FE7FAA000000:7C
5690 55000000AA000000:FF
5698 55000000AAFE00FE:FB
56A0 55000000AA000000:FF
56A8 55000000AA000000:FF
56B0 5500FE7FAA000000:7C
56B8 55000000AA000000:FF
56C0 55000000AAFE00FE:FB
56C8 55000000AA000000:FF
56D0 55000000AA000000:FF
56D8 5500FE7FAA000000:7C
56E0 55000000AA000000:FF
56E8 55000000AAFE00FE:FB
56F0 55000000AA000000:FF
56F8 55000000AA000000:FF
5700 5500FE7FAA000000:7C
5708 55000000AA000000:FF
5710 55000000AAFE00FE:FB
5718 55000000AA000000:FF
5720 55000055AA0000AA:FE
5728 55005555AA00AAAA:FD
5730 55005555AA00AAAA:FD
5738 55005555AA00AAAA:FD
5740 55005555AA00AAAA:FD
5748 55005555AA00AAAA:FD
5750 55005555AA00AAAA:FD
5758 55000055AA0000AA:FE
5760 55000055AA0000AA:FE
5768 55550055AAAA00AA:FD
5770 55550055AAAA00AA:FD
5778 55550055AAAA00AA:FD
5780 55550055AAAA00AA:FD
5788 55550055AAAA00AA:FD
5790 55550055AAAA00AA:FD
5798 55000055AA0000AA:FE
57A0 55000055AA0000AA:FE
57A8 55005555AA00AAAA:FD
57B0 55005555AA00AAAA:FD
57B8 55005555AA00AAAA:FD
```

```
57C0 55005555AA00AAAA:FD
57C8 55005555AA00AAAA:FD
57D0 55005555AA00AAAA:FD
57D8 55005555AA00AAAA:FD
57E0 55005555AA00AAAA:FD
57E8 0000030000000C00:0F
57F0 000030000000C000:F0
57F8 000000030000000C:0F
5800 0081000000810000:02
5808 0081000000810000:02
5810 0081800000818000:02
5818 0081800000818000:02
5820 0081800000818000:02
5828 0081800000818000:02
5830 0081800000818000:02
5838 0081800000818000:02
5840 0081800000818000:02
5848 0081800000810000:82
5850 0081800000810000:82
5858 0081000000810004:06
5860 008100FE00810006:06
5868 0081000700810007:10
5870 008100FF008100AF:B0
5878 008100FF008100AF:B0
5880 008100FF00810006:07
5888 008100FE00810004:04
5890 0081000000810050:52
5898 0081007000810050:C2
58A0 008100000081000D:0F
58A8 0081000200810004:08
58B0 0081000000810000:02
58B8 0081000000810000:02
58C0 0081FF000081AA00:AB
58C8 0081FF000081AA00:AB
58D0 0081AA000081AA00:56
58D8 0081AA000081AA00:56
58E0 0081AA000081AA00:56
58E8 0081AA000081AA00:56
58F0 0081AA000081AA00:56
58F8 0081AA000081AA00:56
5900 0081AA000081AA00:56
5908 0081AA000081AA00:56
5910 0081AA000081AA00:56
5918 0081AA0000000000:2B
5920 0081AA000081AA00:56
5928 0081AA000081AA00:56
5930 0081AA000081AA00:56
5938 0081AA000081AA00:56
5940 0081AA000081AA00:56
5948 0081AA000081AA00:56
5950 0081AA000081AA00:56
5958 0081AA000081AA00:56
5960 0081AA000081AA00:56
5968 0081AA000081AA00:56
5970 DB81FF00FF81FF00:DA
5978 0000000000000000:00
5980 0000000000000000:00
5988 00FF00FF00FF00FF:FC
5990 0000000000000000:00
5998 0000000000000000:00
59A0 FF00FF00FF00FF00:FC
59A8 0000000000000000:00
59B0 0000000000000000:00
59B8 00FF00FF00FF00FF:FC
59C0 0000000000000000:00
59C8 0000000000000000:00
59D0 FF00FF00FF00FF00:FC
59D8 0000000000000000:00
59E0 0000000000000000:00
59E8 0000FF000000FF00:FE
59F0 0000000000000000:00
59F8 0000000000000000:00
5A00 0000FF000000FF00:FE
5A08 FF000000FF000000:FE
5A10 0000000000000000:00
5A18 0000FFFF0000FFFF:FC
5A20 0000000000000000:00
5A28 0000000000000000:00
5A30 00FF00FF00FF00FF:FC
5A38 0000000000000000:00
5A40 0000000000000000:00
5A48 FF000000FF000000:FE
5A50 0000FFFF0000FFFF:FC
5A58 0000000000000000:00
5A60 0000000000000000:00
5A68 00FF00FF00FF00FF:FC
5A70 0000000000000000:00
5A78 0000000000000000:00
5A80 8E00008E48000048:AC
5A88 2B00002B3F00003F:D4
5A90 2B00002B48000048:E6
5A98 8E00008E00000000:1C
5AA0 FF00FFFFFF00FFFF:FA
5AA8 8000000180000001:02
5AB0 80000001CA000001:4C
5AB8 2A0100028F020004:C2
5AC0 9A040008EA080010:A8
5AC8 001000204B200040:DB
5AD0 8840008000800080:48
5AD8 00800080FE830080:01
5AE0 2A8000802A800080:54
5AE8 FE83008000800080:01
5AF0 F881008048810080:42
5AF8 48810080F8810080:42
5B00 0080008000800080:00
5B08 0000000000000000:00
5B10 0000000000000000:00
5B18 0000000000000000:00
5B20 00FFFF0000FFFF00:FC
5B28 0000000000000000:00
5B30 0000000000000000:00
5B38 00FF00FF00FF00FF:FC
5B40 0000000000000000:00
5B48 0000000000000000:00
5B50 FF00FFFFFF00FFFF:FA
5B58 0000000000000000:00
5B60 0000000000000000:00
5B68 00FFFF0000FFFF00:FC
5B70 0000000000000000:00
5B78 0000000000000000:00
5B80 FFFF00FFFFFF00FF:FA
5B88 0000000000000000:00
5B90 0000000000000000:00
5B98 00FFFF0000FFFF00:FC
5BA0 0000000000000000:00
5BA8 0000000000000000:00
5BB0 FF00FFFFFF00FFFF:FA
5BB8 0000000000000000:00
5BC0 0000000000000000:00
5BC8 00FFFF0000FFFF00:FC
5BD0 0000000000000000:00
5BD8 0000000000000000:00
5BE0 0000000000000000:00
5BE8 180018003C003C00:A8
5BF0 3C003C0042004200:FC
5BF8 4200420081008100:86
5C00 8100810081008100:04
5C08 8100810042004200:86
5C10 420042003C003C00:FC
5C18 3C003C0018001800:A8
5C20 0000000000000000:00
5C28 0000000000000000:00
5C30 00180018003C003C:A8
5C38 003C003C00420042:FC
5C40 0042004200810081:86
5C48 0081008100810081:04
5C50 0081008100420042:86
5C58 00420042003C003C:FC
5C60 003C003C00180018:A8
5C68 0000000000000000:00
5C70 0000000000000000:00
5C78 FF00FF00FF00FF00:FC
5C80 0000000000000000:00
5C88 0000000000000000:00
5C90 00FF00FF00FF00FF:FC
5C98 00000055000000AA:FF
5CA0 00000055000000AA:FF
5CA8 FF00FF55FF00FFAA:FB
5CB0 FF00FF55FF00FFAA:FB
5CB8 000000550000FFAA:FE
5CC0 0000005500001CAA:1B
5CC8 003C0055003CFFAA:76
5CD0 00000055000040AA:3F
5CD8 000040550000 5FAA:9E
5CE0 F0005555F000F5AA:29
5CE8 0000F555000055AA:49
5CF0 00005F55000040AA:9E
5CF8 000F4055000F00AA:5D
5D00 00000055000000AA:FF
5D08 0000FF550000FFAA:FD
5D10 F0005555F000AAAA:DE
5D18 000055550000AAAA:FE
5D20 000055550000AAAA:FE
5D28 00F0555500F0AAAA:DE
5D30 000055550000AAAA:FE
5D38 000055550000AAAA:FE
5D40 0F0055550F00AAAA:1C
5D48 000055550000AAAA:FE
5D50 000055550000AAAA:FE
5D58 00FF555500FFAAAA:FC
5D60 0000000000000000:00
5D68 0000000000000000:00
5D70 FFFF00FFFFFF00FF:FA
5D78 00C00000EACB0000:75
5D80 AFC60000EAC30000:22
5D88 AFC6FF00EACBFF00:28
5D90 00C00000C8C10000:49
5D98 28C00000FFCF0000:B6
5DA0 28C000FFC8C100FF:6F
```

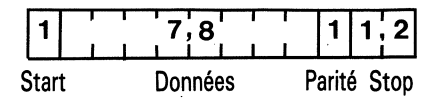
```
5DA8 00C00000FFFF0000:BE
5DB0 FFFF000000000000:FE
5DB8 00FFFF0000FFFF00:FC
5DC0 0000000000000000:00
5DC8 0000000000000000:00
5DD0 FF00FF00FF00FF00:FC
5DD8 0000000000000000:00
5DE0 0000000000000000:00
5DE8 00FFFFFF00FFFFFF:FA
5DF0 0000000000000000:00
5DF8 0000000000000000:00
5E00 FFFF00FF00000000:FD
5E08 0000000000000000:00
5E10 000000000FFFF00:FE
5E18 0000000000000000:00
5E20 000000000FFFFFF:FD
5E28 000055550000AAAA:FE
5E30 000055550000AAAA:FE
5E38 55005555AA00AAAA:FD
5E40 55005555AA00AAAA:FD
5E48 55005555AA00AAAA:FD
5E50 55005555AA00AAAA:FD
5E58 55000000AA000000:FF
5E60 55000000AA000000:FF
5E68 55555500AAAAAA00:FD
5E70 55555500AAAAAA00:FD
5E78 55555500AAAAAA00:FD
5E80 55555500AAAAAA00:FD
5E88 55555500AAAAAA00:FD
5E90 55550000AAAA0000:FE
5E98 55550000AAAA0000:FE
5EA0 55550055AAAA00AA:FD
5EA8 55550055AAAA00AA:FD
5EB0 55550055AAAA00AA:FD
5EB8 55550055AAAA00AA:FD
5EC0 55550055AAAA00AA:FD
5EC8 55550055AAAA00AA:FD
5ED0 55550055AAAA00AA:FD
5ED8 55550055AAAA00AA:FD
5EE0 55550055AAAA00AA:FD
5EE8 55550055AAAA00AA:FD
5EF0 55550055AAAA00AA:FD
5EF8 55550055AAAA00AA:FD
5F00 55550055AAAA00AA:FD
5F08 55550055AAAA00AA:FD
5F10 0000000000000000:00
5F18 0000000000000000:00
5F20 0055555500AAAAAA:FD
5F28 0055555500AAAAAA:FD
5F30 0055555500AAAAAA:FD
5F38 0055555500AAAAAA:FD
5F40 0000000000000000:00
5F48 0000000000000000:00
5F50 55550055AAAA00AA:FD
5F58 55550055AAAA00AA:FD
5F60 55550055AAAA00AA:FD
5F68 55550055AAAA00AA:FD
5F70 55550055AAAA00AA:FD
5F78 55550055AAAA00AA:FD
5F80 55550055AAAA00AA:FD
5F88 55550055AAAA00AA:FD
5F90 55550055AAAA00AA:FD
5F98 55550055AAAA00AA:FD
5FA0 55550055AAAA00AA:FD
5FA8 55550055AAAA00AA:FD
5FB0 55550055AAAA00AA:FD
5FB8 0000000000000000:00
5FC0 0000000000000000:00
5FC8 0000000000000000:00
5FD0 0000000000000000:00
5FD8 0000000000000000:00
5FE0 0000000000000000:00
5FE8 0000000000000000:00
5FF0 0000000000000000:00
5FF8 0000000000000000:00
6000 0000000000000000:00
6008 0000000000000000:00
6010 0000000000000000:00
6018 0000000000000000:00
6020 0000000000000000:00
6028 000000006F6F6F6F:BC
6030 10741D5710700052:CA
6038 10745D5710704052:4A
6040 10743B5710701E52:06
6048 10747B5710705E52:86
6050 78606237001C2504:B6
6058 520757260507412B:4E
6060 0A3784021DDB5002:11
6068 74DB86025DDB5002:61
6070 74DB78605410721D:1A
6078 571074005210725D:0C
6080 571074405284023B:2E
6088 DB500274DB86027B:7F
6090 DB500274DB786054:A8
6098 10723B5710741E52:08
60A0 10727B5710745E52:88
60A8 84021DDB500272DB:1D
60B0 86025DDB500272DB:5F
60B8 78605410701D5710:30
60C0 72005210705D5710:08
60C8 72405284023BDB50:F0
60D0 0272DB86027BDB50:7D
60D8 0272DB7860541070:FB
60E0 3B5710721E521070:04
60E8 7B5710725E528402:8A
60F0 1DDB500270DB8602:1D
60F8 5DDB500270DB7860:AD
6100 5484023BDB500270:B2
6108 DB86027BDB500270:7B
6110 DB7860543778D90C:9B
6118 E4B1840237DB5002:7F
6120 61DB8602FFDB5002:F0
6128 6CDB025F34245726:7D
6130 522F05FDDF374D4D:33
6138 F9F9F9F9F9F9F9F9:C8
6140 F9F9F9F9F9F90031:07
6148 39383500534F4654:E2
6150 F900F5D7D82D0031:FB
6158 333530F500F90000:86
6160 42592ECAD4B6DC00:F9
6168 F962792ECAD4B6DC:32
6170 00D5B3B2C1F9004D:41
6178 4549484F27542E4E:1C
6180 F9F9F9F9F9F9F9F9:C8
6188 F9F9F9F9F9F9007B:51
6190 B6DEDDCADEAFC37D:08
6198 7861150258347860:54
61A0 302F047861A8374D:68
61A8 8402DDDB850261DB:01
61B0 021C3424571061C0:FE
61B8 5224571061C25203:55
61C0 2102AF34125F0220:99
61C8 DBDFDA34DA2F0102:D4
61D0 72DBDFDA34DA2F01:44
61D8 2F152F28374D2846:8D
61E0 2846282325462323:6A
61E8 21AF212523232125:A2
61F0 1D551D551D251A64:A4
61F8 183221701D252125:63
6200 2D741D2521252890:E1
6208 252328232D372846:65
6210 232321551D2321AF:CC
6218 7861987861157860:37
6220 30E436661E7E6350:FF
6228 E436750E7ED90C79:79
6230 621E7866A0840250:D4
6238 DB500262DB8602FF:F1
6240 DB50026CDB025E34:08
6248 245726522F057962:02
6250 B000000000000000:B0
6258 0000000000000000:00
6260 F9F9F9F9F9F9F9F9:C8
6268 F900000000000000:F9
6270 0000000000000000:00
6278 2AD7D82D31333530:CF
6280 2A00000000000000:2A
6288 0000000000000000:00
6290 F9F9F9F9F9F9F9F9:C8
6298 F900000000000000:F9
62A0 0000000000000000:00
62A8 004C4546543A0032:97
62B0 4DFDDF10760002FF:B0
62B8 5210764002FF5210:7B
62C0 761E02FF5210765E:CB
62C8 02FF5210760102FF:DB
62D0 5210764102FF5210:7C
62D8 761F02FF5210765F:CD
62E0 02FF528402FFDB50:03
62E8 0253DB7962F24D4D:97
62F0 F61A24F553107000:FC
62F8 52F61A24F5531070:4E
6300 4052F61A24F55310:1E
6308 701E52F61A24F553:5C
6310 10705E527966B002:C1
6318 19344EFF2F038402:52
6320 16DB500274DB0206:9A
6328 34240200522F0578:58
6330 6580DB500263DB86:D6
6338 0216DB500274DB02:96
6340 0434245726522F05:5F
6348 79637E183CBDE799:EB
6350 106318021A521063:6C
6358 2002165210633902:38
6360 1652840200DB8502:50
6368 6CDB020734240200:AA
6370 522F05106C000203:07
6378 521200796468106C:25
6380 075742750A7E6391:91
6388 740A106C07527964:30
```

```
6390 184D4D106C065742:CD
6398 750A7E63A6740A10:94
63A0 6C0652796418106C:35
63A8 06020052106C0557:32
63B0 42750A7E63BF740A:DF
63B8 106C055279641810:D8
63C0 6C05020052106C04:45
63C8 5742750A7E63D874:45
63D0 0A106C0452796418:D1
63D8 106C04020052106C:50
63E0 035742750A7E62F1:EC
63E8 740A106C03527964:2C
63F0 18106C0302005210:FB
63F8 6C025742750A7E64:68
6400 0A740A106C025279:D1
6408 6418106C02020052:4E
6410 106C0157427962F0:E1
6418 107416574D75007E:31
6420 64788402FFDB5002:8E
6428 73DB8602FFDB5002:02
6430 73DB02FF34245710:0E
6438 643E5202FF75FF26:8F
6440 522F0D02FF344EFF:10
6448 4EFF4D4D2F07106C:99
6450 00574375007E6640:33
6458 106C005274301062:E4
6460 AF52796232000000:0E
6468 10641A0216521062:6A
6470 AF02335279623200:43
6478 E436750E7ED90C74:74
6480 0E75097E64B07409:9B
6488 750D7E64C0740D75:1A
6490 117E64D474117530:F1
6498 7E64EA7430752E7E:91
64A0 65317962F0000000:61
64A8 0000000000000000:00
64B0 E436671529057964:A1
64B8 7800000000000000:78
64C0 10631857437E64CF:D6
64C8 106318527962F002:AA
64D0 017964C810631857:88
64D8 4275207E65977420:E5
64E0 106318527962F04D:F5
64E8 4D4D106339577556:68
64F0 7E6504745675347E:D8
64F8 6513743475747E65:EC
6500 227962F010633902:9B
6508 16524D4D4D4D4D4D:36
6510 7962F01063390256:CF
6518 524D4D4D4D4D4D79:99
6520 62F0106339023452:86
6528 4D4D4D4D4D4D7962:A9
6530 F01063395775347E:1A
6538 654B743475567E65:06
6540 5A745675167E6569:FB
6548 7962F01063390274:ED
6550 524D4D4D4D4D4D79:99
6558 62F0106339023452:86
6560 4D4D4D4D4D4D7962:A9
6568 F010633902565279:BF
6570 62F0000000000000:52
6578 0000000000000000:00
6580 1063395710632052:E8
6588 1063395710641A52:E3
6590 84024A3700000002:09
6598 19106318527962F0:C1
65A0 78D90C0200036CE2:B0
65A8 8202020366E27D02:50
65B0 05344D4D24577430:F2
65B8 26522F078402DFDB:EE
65C0 500265DB8602FFDB:F4
65C8 50026CDB025F3424:52
65D0 5726522F05FDDFE4:C3
65D8 366711290579621A:D1
65E0 F6F6F6F6F6F6F6F6:B0
65E8 F6F6F6F6F6F6F6F6:B0
65F0 F6F6F6F6F6F6F6F6:B0
65F8 0000000053434F52:37
6600 453A003030303538:7C
6608 3230000000000000:62
6610 0000000050555348:40
6618 00454E544552004B:C9
6620 4559000000000000:9E
6628 F6F6F6F6F6F6F6F6:B0
6630 F6F6F6F6F6F6F6F6:B0
6638 F6F6F6F6F6F6F6F6:B0
6640 78D90C840287DB85:CA
6648 0266DB0207342457:FB
6650 1066695224571066:22
6658 6B527866682F10E4:26
6660 B17965A000000000:2F
6668 0300020034125F02:AC
6670 20DBDFDA34DA2F01:F2
6678 0272DBDFDA34DA2F:45
6680 012F153700000000:7C
6688 1864215521553367:02
6690 2846215518640000:60
6698 0000000000000000:00
66A0 106C08020052106C:54
66A8 0902005278D90C37:F1
66B0 676F7E66BB786050:9D
66B8 796317840200DB50:A4
66C0 026CDB0206342402:AB
66C8 09522F05023034E9:DE
66D0 4A789FFF2F064D4D:2F
66D8 8402FFDB850266DB:28
66E0 8602FFDB87026CDB:32
66E8 025F34245726522F:B7
66F0 05FDDFE436670E29:99
66F8 057965A04D4D4D4D:B7
6700 F6F6F6F6F6F6F6F6:B0
6708 F6F6F6F6F6F6F6F6:B0
6710 F6F6F6F6F6F6F6F6:B0
6718 D7D8B031333530A6:CE
6720 00C4B7CFC0A100B5:60
6728 D2C3DEC4B3002121:2C
6730 0000000000005055:A5
6738 534800434C53004B:C8
6740 4559000000000000:9E
6748 F6F6F6F6F6F6F6F6:B0
6750 F6F6F6F6F6F6F6F6:B0
6758 F6F6F6F6F6F6F6F6:B0
```


LA RS-232C EN DOUCEUR

Les micro-ordinateurs utilisent généralement un composant appelé UART (Universal Asynchronous Receiver Transmitter), pour gérer leur voie série. Dans ce cas, les applications développées sur le micro qui utilisent la RS-232C sont relativement simples : on envoie les caractères à l'UART qui se débrouille. D'un autre côté, ce que fait l'UART, le Microprocesseur peut également le faire à l'aide d'un programme adapté. C'est pourquoi, sur les PC-2500, PC-1350 et PC-1450, on ne rencontre pas d'UART.

Dans le principe comment se passe la transmission :
Elle est avant, tout basée sur l'envoi d'un caractère. Celui-ci sera de composé de 7 ou 8 bits de données selon les désirs de l'opérateur. A ces données on ajoutera un bit de départ indiquant que les données arrivent, 1 ou 2 bits d'arrêt indiquant que le caratère est terminé, et au milieu 1 bit de parité ou pas de bit du tout, toujours au choix de l'opérateur. On obtient donc le schéma suivant :



En fonction de paramètres pré-établis on va transmettre des bits d'information avec une cadence précise dépendant de la vitesse choisie. Prenons un exemple : OPEN « 1200,E,7,1 ». Nous aurons : 1 bit de Start, 7 bits de données, 1 bit de parité et 1 bit de stop. Soit la transmission de 10 bits d'information. La vitesse est de 1200 bauds soit 1200

bits par secondes. Pour transmettre un bit il faudra 1/1200 seconde soit 8.33E-04 seconde. Ce caractère sera donc transmis en 8.33 millièmes de seconde.
Côté PC, il va falloir gérer cela et envoyer un bit tous les 8.33E-04 seconde. Pour nous allons devoir travailler au niveau du temps de cycle du microprocesseur. Nos PC utilisent un quartz de 768 KHz, le temps de cycle du MCP est donc de 1/768 000 multiplié ou divisé par une valeur qui s'obtient en regardant le fonctionnement d'une instruction à l'oscilloscope. Cette valeur étant 3 on obtiendra comme temps de cycle 3/768 000 soit 3.90625E-04/3.90625E-06 seconde.

Par conséquent pour transmettre des informations il faudra à 1200 bauds qui il s'écoule 8.33E-04/3.90625E-06 cycles entre deux bits. Le programme devra donc fonctionner en environ 213 cycles et s'articuler ainsi :

- envoi du start bit.
- envoi des 7 bits de données.
- envoi de la parite.
- envoi du bit de stop.

Avant de passer au programme lui-même, il reste à voir ce qu'est la parité. Elle sert à contrôler si le caractère a bien été transmis. Il s'agit ici de transmettre un 1 s'il y a un nombre paire de bit à 0 dans le caractère transmis.
Voyons donc maintenant le programme nécessaire à la transmission d'un caractère en OPEN « 1200,E,7,1 » et ce sans contrôle du flux :

Label	Instruction		Nbre cycle	Commentaires
(SENDD)	EXAB		3	sauve le carac. à émettre dans B
	LP	&0B	2	P pointe sur N le compteur de parite
	ANIM	&00	4	N=0 remise à 0 du compteur
	LIP	&5E	4	P pointe sur le port F où se trouve TD
	ORIM	&08	4	Force le bit 3 à 1 TD=1
	OUTF		3	émission TD à 1 envoi du bit de Start
	WAIT	&08	14	attente de 14 cycles
	LII	&06	4	I=6 longueur−1 du mot à transmettre
(SND3)	WAIT	&A3	169	attente de 169 cycles
	LIP	&5E	4	P pointe sur le port F
	EXAB		3	récupère le caractère à émettre
	SR		2	met dans le carry le bit le plus faible
	EXAB		3	resauve le caractère à émettre
	JRCP	(SND1)	7/4	si bit a 1 on va en SND1
	NOPT		3	attente de 3 cycles
	ORIM	&08	4	TD=1
	OUTF		3	envoi d'un bit de donnée à 1
	JRP	(SND2)	7	on va en SND2
(SND1)	ANIM	&F7	4	TD=0
	OUTF		3	envoi d'un bit de donnée à 0
	INCN		4	N=N+1 compteur du nbre de 0 transmis
	NOPT		3	attente de 3 cycles
(SND2)	DECI		4	I=I−1 compteur de bit à transmettre
	JRNCM	(SND3)	7/4	saut en SND3 si on n'a pas fini le mot
	WAIT	&A6	172	attente de 172 cycles
	LP	&0B	2	P pointe sur le compteur de la parite
	LDM		2	A=(P) dans A on met le nbre de 0 transmis
	LIP	&5E	4	P pointe sur le port F
	TSIA	&01	4	Test du bit 0 à A
	JRNZP	(SND4)	7/4	Saut en SND4 si nbre impair de 0
	NOPT		3	attente de 3 cycles
	ORIM	&08	4	TD=1 car nbre pair de 0 dans le mot
	OUTF		3	émission de la parite à 1
	WAIT	&91	151	attente de 151 cycles
	JRP	(SND5)	7	saut en SND5
(SND4)	ANIM	&F7	4	TD=0 car nbre impair de 0 dans le mot
	OUTF		3	émission de la parite à 0
	WAIT	&98	158	attente de 158 cycles
(SND5)	WAIT	&26	44	attente de 44 cycles
	LIP	&5E	4	P pointe sur le port F
	ANIM	&F7	4	TD=0
	OUTF		3	émission du bit de stop
	WAIT	&CF	213	attente de 212 cycles
	RTN		4	retour au ssp. appelant

Pour utiliser ce programme, il suffit de mettre le caractère à transmettre dans A et d'appeler celui-ci.
Remarques : entre 2 OUTF, il y a toujours 213 cycles de façon à contrôler la vitesse. Les NOPT qui font 3 cycles sont utilisés pour que le prg. soit aussi long quelque soit le chemin pris lors de l'utilisation d'une instruction JRxxx qui fait 7 ou 4 cycles (7=4+3).

Attention :

- la ligne reste à 0 ou à 1 pendant 8.33E-04 secondes.
- le bit de start est un 1.
- les données sont transmises ici sur 7 bits.
- la parite paire est à 1 si le nombre de 0 est pair.
- le bit de stop est un 0.

A l'aide de ce programme et de ces renseignements, écrivez le programme de lecture de données associé à celui-ci. La prochaine fois, nous verrons la réception et nous aborderons le problème du contrôle de la ligne.

I.S

GERER UN MENU.

4 lignes, c'est bien. Mais encore faut-il s'en servir. Présenter un programme, faire des menus... L'intérêt d'un menu est de proposer un choix en fonction duquel on se branchera sur divers sous-programme. Bien, mais comment se brancher sur un SSP ?
Il n'existe pas une méthode, mais une multitude. Voyons en, tout d'abord une propre aux pockets :

Exemple :
10 INPUT « PRG1 » ; A$: GOTO 60
20 INPUT « PRG2 » ; A$: GOTO 70
30 INPUT « PRG3 » ; A$: GOTO 80
40 INPUT « PRG4 » ; A$: GOTO 90
50 GOTO 10
60 CLS : PRINT « PRG1 » : END
70 CLS : PRINT « PRG2 » : END
80 CLS : PRINT « PRG3 » : END
90 CLS : PRINT « PRG4 » : END
Ce programme aura pour effet d'afficher successivement chacun des choix, et de recommençer au 1er si l'on presse ENTER sans avoir pressé au préalable un autre caractère.
En effet lorsque l'on presse ENTER sans avoir mis quoi que ce soit dans l'INPUT, la fin de la ligne sur laquelle celui-ci se trouve est entièrement ignorée, et le contenu de la variable servant à l'INPUT est conservé.
Ainsi, on crée un petit menu qui donne les choix au fur et à mesure.

On peut aussi s'en servir pour faire un choix avec une priorité, en affichant le menu au préalable :
10 CLS : WAIT 0 : PRINT « 1 : PRG1 4 : PRG4 »
20 PRINT « >2 : PRG2 » : PRINT » 3 : PRG3
30 A=2 : INPUT « CHOIX : » : A
40 ON A GOTO 60,70,80,90
50 GOTO 10
60 ...
Ainsi, si l'on presse ENTER directement, on exécutera le programme numéro 2.

Utilisation de l'instruction INKEY$
INKEY$ va nous renvoyer le caractère pressé pendant une scrutation. Pour cette instruction aussi les utilisations sont multiples.
30 PRINT « CHOIX : » ;
40 A=VAL INKEY$: ON A GOTO 60,70,80,90
50 GOTO 40
Dans ce cas, si aucune touche n'est pressée A=0 donc on boucle sans fin. Il en va de même si A²4.
On peut aussi forcer la boucle si aucun caractère n'est pressé.
40 A$=INKEY$: IF A$=« »THEN 40
50 IF A$<« 1 » OR A$>« 4 » THEN 40
60 IF A$=« 1 » CLS : PRINT « PRG1 » : END
70 IF A$=« 2 » CLS : PRINT « PRG2 » : END
80 IF A$=« 3 » CLS : PRINT « PRG3 » : END
90 CLS : PRINT « PRG4 » : END
Le choix de l'instruction à employer est fonction du programme. Généralement on n'utilisera pas 2 fois la même méthode.
Ainsi, pour tester un choix avec des lettres on pourra faire :
10 CLS :WAIT 0 : PRINT « Avant Dernier »
20 PRINT « Bete » : PRINT « Chien »
30 ON ASC INKEY$−840 GOTO 60,70,80,90
40 ... Reprendre l'avant dernière version.
A vous les menus les plus fous.

I.S.



L'AUTOMNE DU 2500

C'est l'automne et les feuilles tombent... du tableur dans le basic...
Le tableur gère ses feuilles comme des tableaux de variables. Ces variables se caractérisent par le symbole@. On rencontre les variables suivantes :
V@$(0,1)★80,X@$(0,0)★16,X@(0,4),Z@(1),B@$(2,1)★12,Z@$(5)★80
W@$(0)★16,U@$(1)★80,U@(4)...
Pour s'en servir sous BASIC, il suffit de les transformer en quelque chose de normal : par exemple changer les@ en A.
C'est ce que fait le programme suivant lorsque l'on fait RUN (pour rétablir l'état normal des variables tableur faire DEF SPACE).

Soit RUN pour l'effet Automne et DEF SPACE pour l'effet Printemps. Le programme vous donne le nom des variables en mémoire, leurs dimensions et leurs longueurs. L'adresse indiquée en tête correspond au début de la mémoire variable.
La variable XA$(,)★16 contient le noms des tableaux et la variable VA$(,)★80 contient les formules.
En reprenant des parties du tableur, il est possible de récupérer des feuilles provenant par exemple d'un 1350 ou d'utiliser le 516P pour imprimer les tableaux.

I.S.

```
10:CLEAR :A= PEEK &6D97+256* PEEK &6D9
   8+1: LPRINT "Adr:";A:W=A-1:X=1
20:IF A>&6C4F BEEP 2: END
30:LPRINT CHR$ PEEK (A-1)+ CHR$ (&40+X
   )+ CHR$ (32+4*(2> PEEK A))+"("+
   STR$ PEEK (A+4)+","+ STR$ PEEK (A+3
   )+")*"+ STR$ PEEK (A+5)
40:POKE A, PEEK A AND &FE OR X:A=A+4+
   PEEK (A+1)*256+ PEEK (A+2): GOTO 20
50:" " POKE &6D97,W-256* INT (W/256),
   INT (W/256):A=W+1:X=0: GOTO 20
```

INITIATION AU LM ESR-H
1ere PARTIE

Devant une demande sans cesse croissante d'explications sur le LM des PC, équipés du microprocesseur SC 61860 (ESR-H), voici un avant propos de ce qui sera je l'espère, une initiation facile à comprendre à tous et à toutes. Vos critiques et suggestions sont évidemment les bienvenues. Au fait, sont concernés tous les PC sauf 1500, 1430, 1246/47. qui n'ont pas d'accès au LM. Les puristes me pardonneront volontiers je pense, quelques abus de langage ; ces articles étant avant tout destinés au débutants, connaissant le Basic. Les lecteurs confirmés peuvent sauter quelques leçons. Vous pourrez aussi consulter avec profit les articles consacrés au Z80 : Le LM des MZ.

Comme certains ont déjà pu le constater, nous avons abondamment parlé de ce LM, en publiant des programmes, dont les commentaires étaient parfois succints. Les plus courageux de nos lecteurs (ou les plus mordus ?) ont donc acquis les différents manuels techniques en anglais, disponibles au Club. Le manuel dédié au PC 1251 convient tout à fait pour les autres machines, hormis les PC 1350 et 2500 qui ont leur propre littérature. Ces manuels qui ne seront pas traduits, sont un guide très précieux et ne font pas double emploi avec cette série d'articles. Ils serviront d'ailleurs, à maintes reprises, de Référence.

Commençons sans plus attendre cette petite mise au point. Prenons un PC au hasard, nous savons qu'il est programmable en Basic. Mais ce que nous ne soupçonnons pas toujours, c'est que la machine ne comprend pas directement ce Langage évolué. Il faut ce que l'on appelle un Interpréteur, qui est un programme en LM (le plus souvent et dans ce cas en mémoire morte). Cet interpréteur est chargé de traduire et contrôler par le MCP (Maitre Contrôle Principal), qui est le programme régissant toute la machine : Calcul, allumage, affichage, etc... Mais comment celà se passe-t-il grosso modo ? Tout d'abord, lors de la saisie d'une ligne en mode programme, la machine (qi ne peut conserver que des valeurs numériques), code les mots clefs avec une valeur appelée : Tooken. Elle la trouve au moyen d'une table, comportant la liste de ces mots réservés, avec, comme information majeure, le nom en ASCII, le tooken, l'adrese du s/pgm LM correspondant.

Les caractères eux sont codés sous forme de code ASCII (sauf pour le PC 1251). Car je rappelle qu'un ordinateur ne peut traiter que des nombres dans un intervalle bien défini : L'ASCII est une normalisation des codes utilisés pour les caractères usuels. Le nombre de codes possibles étant de 256 à la fois, il peut s'avérer (si on a plus de 256 « choses » à coder), que la machine fasse appel à un double codage ; pour les caractères japonais entre autres, mais nous y reviendrons plus tard. Ces codes ne sont en aucun cas, à confondre avec les OP codes (codes machines), qui sont les seuls codes, qui forment les pgms LM exécutables par le microprocesseur. Mais n'allons pas trop vite. Une fois les lignes codées, on passe généralement en mode RUN, et lance le Basic par RUN, GOTO ou DEF label. A ce moment le MCP active l'interpréteur qui cherche le premier code du pgm Basic à exécuter. Si c'est un Tooken, il recherche en table le sous pgm LM correspondant, et récupère si besoin est d'éventuels paramètres. Il décèle aussi les erreurs s'il en trouve, et ce, jusqu'à ce qu'il rencontre un ordre interrompant son travail : à ce moment il rend la main au MCP. Voila en « gros » comment marche le Basic. Nous voyons que tout se passe indirectement, or il est possible de s'adresser directement au microprocesseur, au prix d'une programmation plus ardue, ne laissant pas le droit à l'erreur, mais cmportant des satisfactions tant pour l'esprit que pour l'usage : gain de place, gain énorme de rapidité, des possibilités inexploitables en Basic. Par contre il est illusoire de tout vouloir faire en LM. Nous verrons plus tard, comment exploiter des sous-pgms (routines) touts faits, présents en mémoire morte.

Pour vous prouver qu'au Club nous ne sommes pas des extra-terrestres, nous allons maintenant tenter d'exploiter le vocabulaire un peu ésotérique relatif au langage machine. On confond souvent LM et Assembleur. L'assembleur est en fait un pgm (en LM tout comme l'interpréteur Basic), qui permet de programmer ce LM avec plus de confort. En effet, il permet d'écrire un pgm LM avec une syntaxe bien précise, à l'aide de mots appelés Mnémoniques, car ils signifient quelque chose en anglais : exemple : LIA donne Load Immediate valeur dans Accumulateur, ou SWP donnant SWAP etc... L'assembleur se charge donc de convertir ces mnémoniques en codes machines, suivis ou non d'opérandes (paramètres). Le désassembleur fait l'opération inverse, il traduit des codes machines en mnémoniques. Car il est plus simple de comprendre, écrire et raisonner à l'aide du mot RTN, plutôt qu'avec son code &37. Nous parlons souvent d'adresse, derrière ce mot se cache une notion simple mais importante. En effet, une adresse c'est une valeur comprise entre 0 et 65535 ; elle permet de réperer un octet, tout comme le numéro sur votre porte indique au facteur votre logis. C'est le numéro précis d'un octet, on peut donc en avoir 65536 de suite. Un octet c'est une case — en fait un groupe d'informations élémentaires — refermant une valeur comprise entre 0 et 255. Chaque case (octet) a un numéro d'identification (l'adresse), et renferme une information (code). Les informations élémentaires d'un octet sont appelées bit : binary digit.

Ils sont au nombre de 8 pour un octet (4 pour un quartet=demi-octet). Un octet comprend donc 8 bits. Chaque bit peut prendre 0 ou 1 comme valeur. D'une manière imagée nous avons des cases identifiées par un numéro, comportant 8 compartiments contenant 0 ou 1. Un bit qui veut dire « chiffre binaire » est la



plus petite information possible. Nous voila donc en base 2. Comme le micro ESR-H est un 8 bits, il peut traiter des informations de 8 bits simultanément (octet), et possèdant un bus d'adresses de 16 bits, il peut donc traiter 2^16 = 65536 octets. Nous y reviendrons la prochaine fois, quand nous verrons en détail l'architecture du SC 61860. Voyons maintenant la représentation d'un octet en binaire :

bit no 7 6 5 3 2 1 0
b7 b6 b5 b3 b2 b1 b0

octet
= b7*2^7+b6*2^6+b5*2^5+b4*2^4+b3*2^3+b2*2^2+b1*2^1+b0*2^0
= b7*128+b6*64+b5*32+b4*16+b3*8+b2*4+b1*2+b0*1
valeur max
= 128+64+32+16+8+4+2+1=255
valeur min = 0

En pratique, on travaillera en binaire pour les opérations logiques agissant sur les bits d'un octet.
A propos de base, nous utiliserons la base 16 (hexadécimale), qui est plus commode pour diverses raisons qui s'expliqueront d'elles mêmes, mais aussi parce qu'elle permet de noter la valeur maximale d'un octet (255) sur 2 caractères, et les adresses sur 4. On compte donc ainsi :

DEC 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 etc.
HEX 0 1 2 3 4 5 6 7 8 9 A B C D E F 10 11 etc.

On obtient ainsi 2 quartets que l'on utilise séparément pour les conversions binaires↔HEXA.

Chaque quartet pouvant prendre comme valeur max. 15 = 8+4+2+1, celà donne &F. Donc on a pour un octet de valeur 10010110. Le premier quartet 1001 vaut : 8+1 = 9. Le second 0110 vaut : 4+2 = 6. Donc 10010110 = &96 = 128+16+4+2 = 150. Et le tour est joué.

Vous avez pu aussi remarquer que nous présentions toujours nos listings LM (mêmes désassemblées), sous forme de DUMP. Beaucoup se demandent ce qu'ils doivent en faire. Ce DUMP est la représentation de la mémoire du PC, à partir d'une certaine adresse. Les valeurs sont toutes en héxa. Sur une ligne, de gauche à droite, on trouve une valeur sur 4 caractères : l'adresse. Suivent ensuite accolés deux à deux, séparés, ou complètement soudés, 8 octets ; ils prennent chacun 2 caractères. Eventuellement, il reste une dernière valeur sur 2 car. aussi, qui est la somme des 8 octets précédents. Il permet un contrôle rapide lors de la saisie : c'est le Checksum. Donc, l'adresse à chaque début de ligne varie de 8 en 8, elle correspond au premier octet de la ligne. Ce dump se rentre en mémoire à l'aide d'un utilitaire (il y en a de publiés dans les bulletins) ou grâce à l'instruction POKE adresse, octet1, octet2, etc...

Voilà, c'est tout pour aujourd'hui. La prochaine fois nous ausculterons le microprocesseur, en tachant de comprendre ce que sont les registres, et leur utilité dans les pgms LM.

A bientôt ! !

J.F.V.

SOUS-PGM LM ESR-H

En complément au « TECHNICAL REFERENCE MANUEL », nous vous proposons une première série de sous programmes utilitaires valables pour tous les PC équipés du processeur ESR-H

SOUS-PGM LM	PC 1245/51/55	PC 1260/61	PC 1401/02	PC 1350
LCD ON	11E0	043B	05A2	04B1
LCD OFF	11E5	0437	059E	04AD
STX : BA-->X	11E9	0225	0267	0282
DSTX : BA-1-->X	11F1	022E	0270	0297
STY : BA-->Y	11EE	0222	0264	027D
DSTY : BA-1-->Y	11F5	0242	0278	02B5
BA-->(P+1,P)	11EA	0226	0268	027E
MVXY : Y-->X	1172	1349	15F3	1419
MVYX : X-->Y	1177	1344	15EE	1414
PUSH X	11C7	102E	1321	115C
POP X	11D2	1039	132C	1167
OUT C,A	1FB1	043D	05A4	04B3

LISP

Encore un nouveau langage sur les MZ 700/800, et non des moindres : Le LISP ; notablement amélioré et adapté aux caractéristiques spécifiques des MZ. Pour vous présenter LISP. O : Son auteur (Devinez qui...), Bernard KOKANOSKI et un utilisateur assidu et amoureux de ce langage : Hervé PANETTO.

LISP. O est disponible au club au prix de 150F (cassette, notice et port compris).

INTRODUCTION AU LISP MZ-700

Ce LISP se compose d'un noyau, écrit en langage machine, assez complet puisqu'il reprend la plupart des instructions disponibles sur les quelques rares Lisps disponibles dans le commerce ainsi que celles décrites dans quelques (trop rares, eux aussi) ouvrages sur ce langage trop méconnu des utilisateurs de micro-ordinateurs.

Dans ce noyau, on trouvera les instructions classiques du Lisp plus des ordres spéciaux au MZ-700 permettant de manipuler l'imprimante, en mode texte et graphique, ou de tracer des figures semi-graphiques sur l'écran, en couleur, ainsi qu'un jeu complet de fonctions mathématiques calculant en virgule flottante, en Décimal Codé Binaire, pour plus de précision, et avec, à l'affichage 10 chiffres (13 chiffres internes et calcul par la technique C.O.R.D.I.C avec 16 chiffres pour certaines fonctions mathématiques).

On y trouvera aussi l'instruction SAVE qui permettra à l'utilisateur d'enregistrer le Lisp muni de nouvelles fonctions écrites grâce aux quatre constructeurs DE DF DN DM associés à LAMBDA, FLAMBDA, NLAMBDA et MLAMBDA.

Le spécialiste du langage machine aura à sa disposition POKE, PEEK, CALL et surtout, il aura accès au moniteur de la mémoire vive (le même qu'en Basic) par BYE. Il pourra même, grâce à une toute petite manipulation, réserver une zone de la mémoire, pour y loger un programme en langage machine.

Bernard Kokanosky

LISP : LANGAGE DE L'INTELLIGENCE ARTIFICIELLE

LISP (LIST programming) appelé inconsidérément par les « non adeptes » : Langage Insipide Stupidement Parenthésé est un lagage évolué spécialisé dans la manipulation d'expressions symboliques. **LISP** a été inventé en 1960 par Mc CARTHY mais a, depuis, subi des améliorations et connait aujourd'hui différents dialectes qui bientôt seront peut-être tous réunis autour d'un standard : COMMON LISP.

LISP-0 est une implantation de LISP disponible sur MZ-700 qui ne manque pas d'intérêt et qui permet déjà d'écrire d'assez gros programmes. Cette version est le seul langage sur 700 utilisant toutes les possibilités du MZ (touches de fonctions, RAM 64Ko, graphiques...).

PROGRAMMONS EN LISP

L'objet élémentaire en LISP est l'atome, indivisible, commme son nom l'indique. Un atome peut être une chaîne de caractères ou un nombre.

```
ex : AHHHHHHTOME  /voici un atome
ceci              /un autre
45676             /encore un
```

Une expression symbolique est représentée par un atome ou une liste. Une liste est une suite d'atomes ou de listes entre parenthèses.

```
ex : (ceci EST une LISTE)
()  /une liste vide
NIL /une autre façon de noter une liste vide
((une autre) liste () (PLUS COMPLIQUEE)))
```

Toutes structures données (et principalement les arbres binaires) se ramènent à la construction de listes plus ou moins compliquées.

LISP est un langage dont la particularité qui en fait toute sa puissance est que données et fonctions ont la même syntaxe et peuvent donc indifféremment jouer un rôle ou l'autre. Un interpréteur **LISP** convenable connait au moins 100 primitives pour pouvoir être facilement utilisable mais il suffit d'en connaitre une douzaine pour débuter. Programmer en **LISP** revient donc à augmenter le vocabulaire de base du langage pour créer un environnemment de programmation de plus en plus riche.

Il existe 4 primitives indispensables à la programmation en LISP : CAR, CDR, CONS et LAMBDA qui permettent respectivement de prendre le 1er élément (atome ou liste) d'une liste, prendre le reste d'une liste privée de son 1er élément, placer un élément en début de liste et enfin (le plus important) effectuer du lambda-calcul.

```
ex : ?(CAR' (1 2 3 4) )
     =1
     ?(CDR' (1 2 3 4) )
     =(2 3 4)
     ?(CAR (CDR' (1 2 3 4) ) )
     =2
     ?(CONS '0 ' (1 2 3 4) )
     =(0 1 2 3 4)
```

Le caractère « ' » est un macro-caractère remplaçant la primitive QUOTE qui n'évalue pas son argument.

```
ex : ?(CAR (CDR' (1 2 3 4) ) )
     =2
     ?(CAR' (CDR' (1 2 3 4) ) )
     =CDR
```

La primitive LAMBDA permet de construire des lambda-expressions. Celles-ci sont des fonctions anonymes à usage local.

On peut donner un nom symbolique à ces fonctions en utilisant la primitive 'DE

```
ex : ?(DE CARRE(X)
     (* X X) )
     =CARRE
     ?(CARRE 4)
     =16
```

Vérifions la présence de l'atome LAMBDA dans la définition de la fonction par l'utilisation de

la fonction PRETTY-PRINT.
?(PRETTY-PRINT CARRE)
=(DE CARRE
(LAMBDA (X)
(* X X)))

Un programme quel qu'il soit ne peut pas raisonnablement se concevoir sans tests sur la nature de ses éléments. Ceci es possible en **LISP** grâce à une très grande variété de prédicats et à la primitive COND qui conditionne le programme.

```
ex : ?(DE CHAR(X)
    (COND (EQ X 0)' ZERO
    (EQ X 1)' UN
    (EQ X 2)' DEUX
    T '(NI 0,NI 1,NI 2) ) )
    =CHAR
    ?(CHAR 0)
    =ZERO
    ?(CHAR 1)
    =UN
    ?(CHAR 2)
    =DEUX
    ?(CHAR 5)
    =(NI 0,NI 1,NI 2)
```

La syntaxe de COND est :
COND TEST1 RESULT1
TEST2 RESULT2
....
pour chaque test vérifié le résultat correspondant est évalué. **LISP** est surtout très puissant dans la programmation recursive. La recursivité est la propriété qu'a une fonction de s'appeler elle-même.

Ce genre de fonction permet la décomposition d'un problème en sous-problèmes de plus en plus simples.
Ainsi l'exemple de la fonction factorielle qui peut se définir : n !=n*(n-1) !
0 !=1

```
? (DE FAC(N)
(COND (EQ N 0) 1
T (* N (FAC(SUB1 N) ) ) )
=FAC
```

(SUB1 N) est équivalent à (- 1 N)

```
?(FAC 0)
=1
?(FAC 5)
=120
```

LISP est le langage de la cinquième génération et permet déjà la réalisation de systèmes experts et d'analyseurs syntaxiques. Ce langage sera sûrement celui des traducteurs automatiques qui, pour l'instant ne sont qu'au stade expérimental mais qui ne tarderont pas un jour à être du domaine public : les applications ne manqueront pas.

BIBLIOGRAPHIE

LISP
Patrick Henri WINSTON, Berthold Klaus Paul HORN
Addison-Wesley 1984

LISP, MODE D'EMPLOI
Christian QUEINNEC
Eyrolles 1984

PROGRAMMER EN LISP
Henri FARRENY
Masson 1984
..... et d'autres

SYSTEMES LISP 8 BITS

Sur MZ 700

LISP-0
Un très bon LISP le seul utilisant les caractéristiques spécifiques d'un micro-ordinateur

Sur MZ 700+CPM

P-LISP
Très DECEVANT

LEOLISP 80
Le meilleur LISP 8 bits existant mais introuvable sur le marché (on peut le trouver chez son concepteur à l'INRIA)

H. PANETTO

K-BASIC V.5.5. UNE CORRECTION ET UNE AMELIORATION

Il m'a été récemment signalé que le K-BASIC avait quelques problèmes lors du passage par référence de paramètres chaînes de caractères, uniquement lorsque la procédure ou fonction modifiait la longueur de ce paramètre. Nous allons modifier légèrement le Basic.

Comme d'habitude, charger le Basic et lorsque la sonnerie retentit, faire Shift Break. Grâce à la commande M, changer les octets suivants :

```
IOFD  36  (identificateur K-Basic V.5.6)
5F7D  46
77DD  08    Puis, à partir de 7E4A, entrer les octets :
77E5  4C    0B 0B CD 12 66 13 13 C9
77E6  6C    A partir de 7F44 entrer :
77FA  07    7E 87 4F 06 00 09 23 EB 09 23 EB 4E 23 46 2B 03 03 C3
7806  4A    24 67 23 56 23 71 23 70 23 C9
7807  6C    Enfin, à partir de 7915, on entrera :
7848  CD    4E 23 46 23 E5 69 60 FE 80 D2 44 6D 0I 03 00 D5 EB C5
7849  58    ED BO CI DI EI ED BO C3 0B 67
784A  6D
```

Avec cette correction, les difficultés signalées aux pages 26 et 27 de la documentation, disparaissent totalement !

Nous allons profiter de cette correction du Basic, pour lui ajouter deux instructions DEEK et DOKE dont le fonctionnement est presque

identique à PEEK et POKE, mais qui travaillent sur deux octets consécutifs : Par exemple DOKE §F000,§A901 placera 01 dans la mémoire n° F000 et A9 dans la mémoire F001. Inversement PRINT DEEK(§F000) donnera 43265 (soit §A901).

Pour DEEK

Nous lui donnons le code FF A9.

Il faut d'abord décaler la troisième table des mots-clés du Basic. Pour cela, grâce à la commande M, nous écrivons, à partir de 9000 :11 F5 81 21 F2 81 01 5F 00 ED B8 C3 AD 00.

Et nous faisons J9000 qui réalise le décalage.

Puis nous écrivons dans les octets 8193 et suivants : 44 45 45 CB.

L'adresse de la routine sera 6C3E qui sera placée dans les octets

844F 3E
8450 6C

Nous écrivons alors la routine à partir de 7E3E : 21 A1 72 CD 5B 36 5E 23 56 C3 26 51.

Pour DOKE

Ce mot-clé doit se trouver avant DO dans la première table. J'ai pensé lui donner le code 9A, celui de CONT et de donner à CONT le code CD. Cette transformation n'est pas gênante, puisque CONT n'est utilisé qu'au clavier et non dans un programme.

Il faudra donc décaler la première table des mots-clé. On modifiera légèrement le programme de transfert situé à 9000 en écrivant dans les octets

9001 F4
9002 7C
9004 F1
9005 7C
9007 9D

On fera ensuite J9000 ce qui réalise le transfert.

On écrit ensuite dans

7C54 43
7C55 4F
7C56 4E
7C57 D4

et dans

7B88 44
7B89 4F
7B8A 4B
7B8B C5

Ensuite on place dans

8313 23
8314 6C

et dans

8379 60
837A 20

Il reste à écrire la routine en plaçant, en mémoire, à partir de 7E23 les octets : CD 50 48 CD E4 67 D5 CD 50 48 E3 73 23 72 23 EB EI CD 42 4F FE 2C CO 23 C3 29 6C et à ré-enregistrer le Basic.

On remarquera que la syntaxe DOKE adresse, valeur, valeur, valeur... est acceptée.

Après toutes ces modifications, les zones libres du Basic sont :

Zone 1 : de 6AF5 à 6C22 (302 octets)
Zone 2 : de 6D41 à 6D43 (3 octets !)
Zone 3 : de 6FF6 à 6FFC (7 octets)

UNE PREMIERE UTILISATION

Lors de l'utilisation de CTRL L pour lister les variables sur l'imprimante, il se peut qu'il soit inutile de lister les tableaux, ce qui consomme beaucoup de papier. On fera DOKE §6DC9,§6DD5 et la routine sautera les tableaux. DOKE §6DC9,§6C52 redonnera le fonctionnement normal.

B.Kokanoski

FORTH

LE COIN DU FORTH... NE... !

Me revoici, pour vous parler à nouveau de votre langage préféré... Forth, ou plutôt d'*Ediforth*, son implantation chère à la famille MZ.

Dans mon précédent article, où entre nous il était bien difficile de trouver un seul défaut au langage *Ediforth*, ses principaux concepts étaient détaillés. Avant de vous livrer aux joies de l'interprétation et de la compilation de mots nouveaux sous *Ediforth*, je vous propose une brève révision.

Machine Forth

Forth est un langage dont le fonctionnement ressemble à celui d'un microprocesseur 16 bits qui serait doté d'instructions comparables en puissance, à celles d'un langage évolué tel que Basic, ou bien encore Pascal. D'ailleurs, un microprocesseur dédié à Forth a été mis récemment sur le marché.

Mots Forth

Les mots Forth qui sont l'équivalent des instructions des autres langages, sont construits, soit en faisant appel à d'autres mots Forth, il s'agit alors de mots de haut niveau, soit à l'aide de routines en langage machine, on parle dans ce cas de mots de base ou primitives.

De ce fait, programmer en Forth consiste simplement à construire de nouveaux mots, à l'aide des mots déjà existants.

Tous les mots de votre version d'*Ediforth* peuvent être visualisés en tapant le mot "VLIST", ou imprimés en frappant la séquence suivante : "PRTON VLIST CR PRTOFF"

Ce bel ensemble de mots que vous venez d'obtenir, par les manipulations ci-dessus, tous moins parlant les uns que les autres, je vous l'accorde et que vous allez apprendre à découvrir dès aujourd'hui, forme le dictionnaire d'*Ediforth*.

Piles Forth

L'un des moyens utilisés par les mots Forth pour se passer de l'information, caractères, nombres ou adresses, ceci étant affaire de convention, c'est la pile de données. Son fonctionnement est en tous points comparable à la fameuse pile d'assiettes, du plus haut placard de votre cuisine... !

Tous les mots Forth qui entrent dans la composition d'un autre mot, sont traités par la machine Forth comme autant de sous programmes à exécuter. Celle-ci stocke donc les adresses de retour correspondantes, sur une autre pile : la pile des retours.

Modes de fonctionnement de Forth

Lorsque Forth exécute un mot ou une suite de mots, entrés au clavier ou à partir d'un péri-

phérique, il est en mode interprétation. Par contre, lorsqu'il crée la définition d'un nouveau mot dans son dictionnaire, il est en mode compilation.

Standardisation de Forth

Les implantations de Forth qui répondent au standard Fig-Forth ont un noyau d'environ 220 mots communs, ayant les mêmes noms et un fonctionnement identique. C'est le cas des différentes versions d'**Ediforth** dans lesquelles seuls quelques mots inutiles ont été supprimés. Une application, bâtie uniquement à l'aide de mots de ce noyau, est réputée pouvoir fonctionner sur n'importe quel ordinateur, doté d'une implantation Forth conforme au standard.

Après ce bref rappel des principaux concepts de Forth...

DES MOTS POUR DEBUTER EN FORTH

Avant de livrer à votre curiosité fébrile un premier vocabulaire de 16 mots, pour débuter en Forth, étudions tout d'abord les conventions de notation utilisées, dans les descriptions qui les accompagnent.

<A---B> = état de la pile :

A -> avant exécution du mot

B -> après exécution du mot

<A1,A2,A3 --- B1,B2> = cas où plusieurs objets sont présents avant et après l'exécution du mot. A3 et B1 sont les objets au sommet de la pile.

Les objets Ax et Bx peuvent prendre les différentes formes ci-dessous :

- n = nombre entier simple longueur compris entre – 32768 et + 32767 sur deux octets.
- d = nombre double longueur compris entre – 2147483648 et + 2147483647 sur quatre octets.
- adr = adresse mémoire sur deux octets.
- f = drapeau résultant d'une opération logique ou d'un test avec 1 ou + = vrai et 0 = faux.

Voici vos 16 premiers mots **Ediforth** tant attendu et rendez-vous un peu plus loin, pour commencer à les utiliser et en créer de nouveaux... !

*

<n1,n2 --- n> laisse sur la pile le produit signé simple longueur "n" des deux nombres simple longueur "n1" et "n2".

+

<n1,n2 --- n> laisse sur la pile la somme "n" des nombres "n1 et n2".

-

<n1,n2 --- n> met sur la pile la valeur "n" qui est la différence de "n1 – n2".

/

<n1,n2 --- n> laisse sur la pile le quotient signé des nombres simple longueur "n1 et n2".

•

<n ---> affiche, sur le périphérique de sortie, le nombre simple longueur signé "n", en fonction de la valeur de la variable utilisateur "Base".

•"

nombres "n1 et n2".

-

<n1,n2 --- n> met sur la pile la valeur "n" qui est la différence de "n1 – n2".

/

<n1,n2 --- n> laisse sur la pile le quotient signé des nombres simple longueur "n1 et n2".

•

<n ---> affiche, sur le périphérique de sortie, le nombre simple longueur signé "n", en fonction de la valeur de la variable utilisateur "Base".

•"

<---> affiche, sur le périphérique de sortie, la chaîne de caractères qui le suit. Celle-ci se termine obligatoirement par le caractère. Avec Ediforth la longueur maximum de la chaîne est égale à celle de l'écran.

!

<n1,adr ---> range la valeur "n" aux adresses "adr" et "adr + 1".

<n,adr --- n> laisse sur la pile le nombre simple longueur "n" situé aux adresses "adr et adr + 1".

:

<---> s'utilise sous la forme : ": NNNN ...... ;" Ce mot marque le début de la définition d'un nouveau mot "NNNN" qui se termine par le mot ";" qui en achève la compilation et finit de le ranger dans le dictionnaire.

Base

<--- adr> variable utilisateur contenant la base numérique courante utilisée lors des opérations de conversion numérique des entrées/sorties.

CR

<---> réalise un retour charriot et un saut à la ligne, sur le périphérique de sortie (écran ou imprimante).

DO

<n1,n2 ---> lors de l'exécution.

<--- adr,n> lors de la compilation.

Ne peut s'utiliser que dans une définition, avec l'une des formes suivants :

"DO ... LOOP"

"DO ... +LOOP"

En mode d'exécution "DO" marque le début d'une séquence à exécuter "n1 – n2" fois. Il s'agit d'une structure de boucle comparable au "FOR ... NEXT" du BASIC, où l'indice de boucle inférieur est égal à "n2" et le supérieur à "n1 – 1".

Par exemple une boucle "10 1 DO ... LOOP" sera parcourue neuf fois (de 1 à 9).

Lors de la compilation, au sein d'une définition, le mot "DO" compile la procédure d'exécution "(DO)" à sa place. Il dépose sur la pile l'adresse suivante "adr" ainsi qu'un nombre simple longueur "n", pour faciliter la vérification de la présence de "LOOP ou +LOOP" et indiquer l'adresse de début de boucle à ces derniers. (Voir aussi le mot "?Pairs").

Le mot "DO" utilise les mots qui suivent pour contrôler le fonctionnement de la boucle :

- "LOOP et +LOOP" pour marquer la fin de la boucle et effectuer un branchement au mot qui suit "DO" s'il y a lieu.
- "I" pour mettre l'indice courant de la boucle sur la pile.
- "LEAVE" pour sortir de la boucle avant son achèvement, suite à un test.

Emit

<n---> affiche, sur le périphérique de sortie sélectionné (écran et/ou imprimante, voir aussi les mots "PRTON et PRTOFF"), le caractère ASCII dont le code "n" est sur la pile.

Le contenu de la variable utilisateur "OUT" est incrémenté à chaque utilisation du mot "EMIT".

Key

<--- n> attend l'enfoncement d'une touche au clavier et en laisse le code ASCII "n" sur la pile.

LOOP

<adr,n ---> en mode compilation

Ne peut s'utiliser que dans une définition, avec la forme suivante :

"DO ... LOOP"

En mode exécution, "LOOP" marque la fin d'une séquence répétitive et réalise un saut en son début, si le nombre de répétitions prévu n'est pas atteint.

Lors de la compilation, au sein d'une définition, le mot "LOOP" force la compilation de la procédure d'exécution "(LOOP)", à sa place.

L'adresse "adr" sert à calculer le déplacement à réaliser, pour effectuer un branchement au mot suivant "DO". Le nombre simple longueur "n" permet de vérifier l'existence, préalable, du mot "DO".
(Voir aussi les mots "DO, I, LEAVE et +LOOP").

Slook
<---> affiche le contenu de la pile sans le modifier. Ce mot est un outil très utile, pour la mise au point de la définition d'un autre

VOICI LES TRAVAUX FORTH... ES !

(Pourra-t-il continuer ainsi jusqu'à la fin de l'article ? N.D.L.R.)
Je vous propose d'essayer de faire travailler l'interpréteur d'**Ediforth**, avec les différents mots que nous venons de voir et de terminer en créant de nouveaux mots, avec des noms en breton, si vous le désirez... !

Jouons avec les changements de base :

• 10 1000 2 BASE ! . BASE ! CR
1111101000
<OK>
•

Attention Forth ne connaît qu'un seul séparateur : l'espace.
Nous avons mis successivement, sur la pile, les nombrfes 10,1000 et 2. Ce dernier nombre est situé au sommet de la pile. Vous pouvez le vérifier, aisément, en intercalant le mot "SLOOK" entre 2 et BASE. Il est ensuite chargé dans la variable "BASE", à l'aide du mot "!". Dès cet instant, les entrées/sorties d'**Ediforth** se réalisent en binaire (= base 2).

Le mot "." affiche le nouveau sommet de la pile (= 1000 en décimal), en binaire et l'efface de celle-ci par la même occasion. Là aussi, une utilisaton judicieuse du mot "SLOOK", vous permettra de vérifier l'évolution de la pile. Nous terminons ce premier essai d'**Ediforth**, en revenant en base décimal par chargement de la valeur 10, mise initialement sur la pile, dans la variable "BASE".

Nous allons maintenant, enfin, créer notre premier mot en auvergnat... :

```
  FOUCHTRA 6 1 DO
             KEY EMIT
             LOOP CR ;
<OK>
•
```

Comme son nom ne l'indique pas, ce mot permet la saisie et l'affichage d'un mot de cinq lettres... sur l'écran de votre MZ.

Créer un mot nouveau, c'est facile avec **Ediforth**. On indique tout d'abord le début de la définition avec":" ; suivi immédiatement après, avec toutefois au moins un espace de séparation, du nom du mot qe vous désirez créer, "FOUCHTRA" en l'occurence. Ensuite il reste à écrire les mots et/ou les nombres qui composent la définition proprement dite et à terminer celle-ci avec";".

Voilà, c'est fait, vous venez de créer votre premier mot sous **Ediforth**. Une pause café paraît nécessaire... !
• : PAUSE-CAFE. "Pause café, un quart d'heure de détente" CR ;
<OK>
•
Je vous laisse deviner ce que fait ce dernier mot et revenons maintenant au fonctionnement du mot "FOUCHTRA".
Celui-ci est composé d'une boucle "DO ... LOOP" dont le fonctionnement se rapproche du "FOR... NEXT" du BASIC. Les bornes de la boucle sont dans l'ordre inverse, par rapport à ce dernier et la borne de fin est toujours supérieure de un au nombre d'itérations. Ainsi "6 1" comme bornes, réalisent l'exécution, cinq fois de suite, des mots compris entre "DO ... LOOP".

Ces mots, justement, que font-ils ?
Le mot "KEY" attend la frappe d'une touche au clavier et dépose le code ASCII correspondant sur la pile.
Son complément naturel "EMIT" prend le code ASCII sur la pile, l'affiche à l'emplacement courant du curseur et avance ce dernier d'une position.

Agitez en tournant cinq fois de suite et voilà votre mot de cinq lettres qui s'affiche miraculeusement.
Si cinq lettres ne vous suffisent pas pour vous exprimer, alors il vous faut augmenter la borne haute de la bouche... !

Pour terminer, encore une petite manipulation ? Alors allons-y pour quelques opérations de lectures/écritures en mémoire.

Le "PEEK(adr)" du BASIC se dit "adr " en Forth, avec "adr" l'adresse à lire. Le mot " " a toutefois la particularité de lire deux octets à la fois, soit les adresses "adr et adr+1". En effet, n'oubliez pas qu'**Ediforth** se comporte comme un super microprocesseur 16 bits.

Heureusement, il existe un mot "C " qui ne lit qu'un octet à la fois, ouf !
Toujours dans le même ordre d'idée, le "POKE adr,n" du BASIC, se dit "n adr!" en Forth et écrit le nombre simple longueur "n" aux adresses "adr et adr+1".
Mais là aussi, nous disposons du mot "C!" qui n'écrit qu'un octet à la fois.

Si malgré tout ce que nous venons de voir, vous êtes toujours un incorrigible fervent de BASIC, voici la botte secrète d'**Ediforth** :
• : PEEK C ;
: POKE C! ;
<OK>
Une seule devise pour **Ediforth** :
Vous changez de langage...
... changez de mots, si vous le désirez... !

D. BEURRIER



ZEN-DISK

Si vous possédez le mini-DOS dont nous avons parlé dans le bulletin n° 14 et l'assembleur ZEN X, je vous propose de vous affranchir du lecteur de cassettes au profit du lecteur de disquetes. En effet, le ZEN X, ne prévoit la sauvegarde de vos précieux travaux en langage machine que sur le périphérique lecteur de cassettes. Aussi, à l'aide de 3 commande (« I », « O » et « V ») détournées du ZEN X d'origine et avec un peu de patience, l'accès de vos programmes Sources, leurs combinaisons, le chargement des programmes assemblés et leur exécution se feront directement vers ou à partir de vos lecteur de disquettes.

Confection du programme ZEN-XP-FD :

Tout d'abord, entrez en mémoire :
• le DUMP N° 1 (par la méthode de votre choix)

Ceci fait :
1. Charger sur la cassette le programme ZEN-FD que vous venez de créer :

• par *S2C0031FF00AD et NAME « ZEN-FD ».

2. Charger le ZEN X d'origine et répondre ZEN G0+CR+CR pour revenir, au moniteur

3. Recharger ZEN-FD par la commande L du moniteur.

4. Modifier les octets suivants :
M18C2+CR
18C2 C5 C3+CR
18C3 2E 66+CR
18C4 96 2D+CR
18C5 CD Shift Break

M 18 CC
18CC ED C3
18CD 78 A5
18CE F5 2F
18CF CD Shift Break

M1603
1603 CD C3
1604 BB 4A
1605 15 30
1606 FD Shift Break

M138A
138A 00 00
138B 30 32
138C 00 00
138D 30 32
138E 00 Shift Break

5. Sauvegarder sur cassette le ZEN-XP-FD par la commande *S120031FF00AD

Création de la disquette ZEN-XP-FD

1. Prendre une disquette neuve que vous initialiserez par « I » et « S » du programme UTILITY (ou MASTERCOPY) revenir au moniteur par « M ».

2. Recharger le ZEN-XP-FD de la cassette réenroulée.

3. Faire JF57E puis W1,0030,2000,1200+CR

4. Charger en mémoire par
R1,0000,0100,1200+CR
puis R1,000F,0100,1300+CR
revenir alors au moniteur par « M »+CR.

5. Entrer alors le DUMP N° 2 suivant puis faire :
M 1306+CR
1306 00 FF+CR
1307 00 FF+CR
1308 00 FF+CR
1309 00 FF+CR
130A 00 Shift break

6. Après retour au MINI-DOS par JF 57E lancer la commande d'écriture
W1,0000,0100,1200+CR puis
W1,000F,0100,1300+CR revenir enfin au moniteur par M+CR.

Mode d'emploi du ZEN-XP-FD

Charger tout d'abord le ZEN-XP-FD par la commande F du moniteur.

1. La commande 0 de sortie du MZ-700 vers la disquette

a) Pour sauvegarder sur disquette un programme (ou listing) Source que vous récupérez d'une cassette ou que vous venez d'écrire exécutez les commandes suivantes :

ZEN>0+CR
DATA>4+CR
NAME>nom (du listing)+CR

b) Si après assemblage correct, vous désirez sauvegarder le programme Objet ainsi généré, exécutez les commandes :

ZEN>0+CR
DATA>1+CR
NAME>nom (du programme)+CR
START>l'adresse de début du programme (suivant la syntaxe du ZEN X)+CR.
STOP>l'adresse de fin de programme+CR
EXEC>l'adresse de démarrage du programme+CR

c) la commande PLOT ON du BASIC sera réalisée avec le ZEN-XP-FD par les instructions suivantes :
ZEN>0+CR

DATA>9+CR (seuls, les messages d'erreur resteront à l'écran).

d) La commande PLOT OFF est réalisée par :
ZEN>0+CR
DATA>0+CR

2. La commande V de visualisation du contenu de la disquette ZEN>V+CR vous donnera le contenu de votre disquette.

Les nombres indiqués sont en hexa
Les codes sont les suivants :
T. signifie type de programme :
1. Pour les programmes en langage machine
3. Pour les fichiers à accès séquentiel
4. Pour les sources du ZEN ou pour les fichiers à accès direct
5. Pour les programmes en BASIC. Pour le ZEN-XP-FD, seuls les types 1 et 4 (listings sources) sont utilisables.

Adem : c'est l'adresse d'exécution du programme objet
Long : c'est la longueur en octets du programme
Asec : c'est le numéo du secteur de la disquette à partir duquel est chargé le programme.
Sur le message « Appuyez ↦ CR » en plus de cette possibilité, vous pouvez faire shift break, ce qui interromp la recherche et fait revenir au ZEN.

3. La commande I de chargement du MZ-700 à prtir de la disquette
Après ZEN>I+CR, au message NAME répondre le nom du Source ou de l'objet que vous désirez respectivement transférer ou exécuter en mémoire centrale.

REMARQUE :

Lors de la visualisation par V, si un ou des espaces apparaissent devant le nom du programme, il faudra le charger sans omettre (après NAME>)ces espaces sinon le message « BADE READ » s'affichera.
Par ailleurs s'il n'y a pas ambiguité, vous pouvez vous contenter de la ou des premiers caractères identifiant le nom du source ou de l'objet que vous voulez charger.

Autres remarques :

• Les programmes ou les sources peuvent très bien être chargés sur une autre disquette.

• La disquette ZEN-XP-FD est compatible avec les utilitaires comme Fiching CMT, Utility ou Mastercopy.

Xavier PIERSON

DUMP N° 1

```
2D57 0D 53 A5 A6 36 32 20 2D  .Suite -
2D5F 2D 2D 3E 20 43 52 0D CD  --> CR.CD
2D67 D8 2D 2E 36 CD B0 1A 7D  D8-.6CDB0.7D
2D6F FE 01 28 1B FE 04 28 17  FE.(.FE.(.
2D77 FE 09 28 08 FE 00 28 02  FE (.FE.(.
2D7F 18 E5 3E 56 21 DF 13 77  .E5>V!DF.77
2D87 21 F8 13 77 C3 00 12 11  !F8 77C3.
2D8F 00 2C 12 CD B3 2D 32 2E  .,.CDm-2.
2D97 2D 3A 00 2C FE 04 28 25  -:.,FE.(%
2D9F CD CB 1A 22 14 2C ED 53  CDCB.".,EDS
2DA7 16 2C 2E 8A CD B6 1A 22  .,.8ACDB6."
2DAF 18 2C 18 31 2E 71 CD E2  .,.1.q CDE2
2DB7 19 73 FE 10 30 F6 13 21  .sFE.0F6.!
2DBF 0E 13 ED B0 C3 2A 8A 13  ..EDnC3*8A.
2DC7 22 16 2C EB 2A 8C 13 ED  "..EB*8C.ED
2DCF 52 22 14 2C 21 00 12 18  R"..!..
2DD7 D6 21 01 2C 06 11 3E 0D  D6!.,..>.
2DDF 77 23 10 FC C3 2A 14 2C  77#.FCC3*.,
2DE7 7D FE 00 28 01 24 7C 32  7DFE.(.$7C2
2DEF 28 2D CD 37 2E DA CD 2F  (-CD7.DACD/
2DF7 CD B1 2E DA 14 2E CD 5E  CDB1.DA..CD^
2DFF 2F ED 5B 1E 2C ED 4B 14  /ED[.,EDK.
2E07 2C 2A 16 2C CD 1A 2E CD  ,*.,CD..CD
2E0F 5D F0 C3 00 12 CD 5D F0  ]F0C3..CD]F0
2E17 C3 C6 20 DD 21 08 10 DD  C3C6 DD!..DD
2E1F 36 00 00 DD 73 01 DD 72  6..DD73.DD72
2E27 02 DD 71 03 DD 70 04 DD  .DD71.DD70.DD
2E2F 75 05 DD 74 06 C3 54 F1  75.DD74.C3TF1
2E37 11 0F 00 CD A8 2E CD 5D  ...CD3.CD]
2E3F F0 3A 21 2C 6F 26 00 11  F0:!,oF&..
2E47 25 2C 0E 00 13 1A 06 08  %,......
2E4F CB 47 20 03 CB 41 20 0D  CBG .CBA .
2E57 CD 76 2E 18 08 CB 41 28  CD76...CBA(
2E5F 04 CD 87 2E D0 23 D5 E5  .CD87.D0#D5E5
2E67 EB 21 61 04 ED 52 E1 D1  EB!61.EDRE1D1
2E6F 28 2C 0F 10 DB 18 D5 22  (,..DB.D5"
2E77 1E 2C ED 53 26 2D 0C F5  .,EDS&-.F5
2E7F 3E 08 90 32 29 2D F1 C9  >.902)-F1C9
2E87 E5 D5 ED 5B 1E 2C ED 52  E5D5ED[.,EDR
2E8F 0C 16 00 F5 3A 28 2D 5F  ...F5:(-_
2E97 F1 ED 52 D1 E1 C9 2B CB  F1EDRD1E1C9+CB
2E9F 41 28 04 CD 87 2E D0 3F  A(.CD87.D0?
2EA7 C9 01 00 01 21 20 2C C3  C9...! ,C3
2EAF 2D 30 06 08 AF 32 22 2D  -0..j2"-
2EB7 11 0F 00 13 ED 53 2A 2D  ....EDS*-
2EBF ED 43 2C 2D CD A8 2E ED  EDC,-CD3.ED
2EC7 5B 2A 2D ED 4B 2C 2D CD  [*-EDK,-CD
2ECF 11 2F D8 10 E6 3A 2E 2D  ./D8.E6:.-
2ED7 FE 00 C8 3A 22 2D FE 02  FE.C8:"-FE.
2EDF 28 01 C9 ED 5B 20 2D CD  (.C9ED[ -CD
2EE7 A8 2E 3A 23 2D 21 20 2C  3.:#-! ,
2EEF 11 20 00 47 FE 00 28 03  . .GFE.(.
2EF7 19 10 FD EB 21 00 2C 01  ..FDEB!.,.
2EFF 20 00 ED B0 01 00 01 21   .EDn...!
2F07 20 2C ED 5B 20 2D C3 1A   ,ED[ -C3.
2F0F 2E C9 C5 D5 21 20 2C 06  .C9C5D5! ,.
2F17 08 11 01 2C 7E 22 24 2D  ...,7E"$-
2F1F FE 00 28 18 23 0E 11 1A  FE.(.#...
2F27 BE 20 28 23 13 1A FE 0D  { (#..FE.
2F2F 28 06 0D 79 FE 00 20 EF  (..79FE. EF
2F37 D1 C1 3F C9 3A 22 2D FE  D1C1?C9:"-FE
2F3F 02 28 10 3C 32 22 2D D1  .(.<2"-D1
2F47 ED 53 20 2D D5 3E 08 90  EDS -D5>.90
2F4F 32 23 2D 2A 24 2D 11 20  2#-*$-.
2F57 00 19 10 BD D1 C1 C9 11  ...4D1C1C9.
2F5F 0F 00 CD A8 2E ED 5B 22  ..CD3.ED["
2F67 2C 3A 28 2D 6F 26 00 19  ,:(-oF&..
2F6F 22 22 2C ED 5B 26 2D 67  "".ED[&-67
2F77 3A 29 2D 84 67 06 08 1A  :)-8467...
2F7F CB C7 25 F5 7C FE 00 28  CBC7%F57CFE.(
2F87 08 F1 0F 10 F3 12 13 18  .F1..F3...
2F8F EC 3E 08 80 47 F1 0F 10  EC>..GF1..
2F97 FD 12 11 0F 00 01 00 01  FD......
2F9F 21 20 2C C3 1A 2E CD D8  ! ,C3..CDD8
2FA7 2D 11 00 2C CD B3 2D AF  -...CDm-j
2FAF 32 2E 2D CD B1 2E 30 16  2.-CDB1.0.
2FB7 2A 24 2D 11 00 2C 01 20  *$-...,.
2FBF 00 ED B0 3A 00 2C FE 04  .EDn:.,FE.
2FC7 28 0A FE 01 28 25 CD 5D  (.FE.(%CD]
2FCF F0 C3 17 16 ED 5B 1E 2C  F0C3..ED[.,
2FD7 ED 4B 14 2C 2A 8C 13 CD  EDK.,*8C.CD
2FDF 2D 30 CD 5D F0 2A 8C 13  -0CD]F0*8C.
2FE7 ED 5B 14 2C 19 22 8C 13  ED[.,."8C.
2FEF C3 00 12 2A 18 2C 7C FE  C3..*.,7CFE
2FF7 A0 30 27 31 00 CF 11 00  q0'1.00...
2FFF CF E5 21 0C 30 01 40 00  .E5!.0.@.
3007 ED B0 DA 00 CF ED 5B 1E  EDnDA..ED[.
300F 2C ED 4B 14 2C 2A 16 2C  ,EDK.,*.,
3017 D2 26 30 CD 21 CF CD 5D  D2&0CD!.CD]
301F F0 C3 11 0C 30 18 DA CD  F0C3..0.DACD
3027 2D 30 CD 5D F0 C9 DD 21  -0CD]F0C9DD!
302F 08 10 DD 36 00 00 DD 73  ..DD6..DD73
3037 01 DD 72 02 DD 71 03 DD  .DD72.DD71.DD
303F 70 04 DD 75 05 DD 74 06  70.DD75.DD74.
3047 C3 B5 F0 AF 32 A3 31 11  C3B5F0j2w1.
304F 0F 00 CD A8 2E ED 5B 22  ..CD3.ED["
3057 2C 2A 24 2C ED 52 11 9E  ,*$,EDR.p
305F 31 CD 2D 31 11 7C 31 CD  1CD-1.7C1CD
3067 39 31 CD 06 00 11 A4 31  91CD...s1
306F CD 39 31 11 CD 31 CD 39  CD91.CD1CD9
3077 31 11 A4 31 CD 39 31 CD  1.s1CD91CD
307F A1 30 11 10 00 ED 53 F6  a0...EDSF6
3087 31 CD A8 2E CD B7 30 ED  1CD3.CDn0ED
308F 5B F6 31 13 ED 53 F8 31  [F61.EDSF81
3097 21 18 00 ED 52 CA B1 30  !..EDRCAB10
309F 18 E7 11 00 00 3A 21 2C  .E7...:!,
30A7 FE 30 C8 CD A8 2E CD B7  FE0C8CD3.CDn
30AF 30 C9 CD 5D F0 C3 00 12  0C9CD]F0C3..
30B7 21 20 2C 06 08 11 2F 2D  ! ,...-/-
30BF E5 7E FE 00 28 38 FE 80  E57EFE.(8FE.
30C7 28 34 E5 C5 CD 71 31 C1  (4E5C5CD71 1C1
30CF E1 E5 CD 21 31 13 23 CD  E1E5CD!1.#CD
30D7 0F 31 DD E1 11 14 00 DD  .1DDE1...DD
30DF 19 11 43 2D CD 06 31 CD  ..C-CD.1CD
30E7 06 31 CD 06 31 DD 23 DD  .1CD.1DD#DD
30EF 23 DD 23 DD 23 CD 27 31  #DD#DD#CD'1
30F7 CD 36 31 CD 67 09 E1 11  CD61CD67.E1.
30FF 20 00 ED 5A 10 B7 C9 CD   .EDZ.oC9CD
3107 27 31 DD 23 DD 23 13 C9  '1DD#DD#.C9
310F 7E FE 0D C8 12 13 23 18  7EFE.C8..#.
3117 F7 F5 0F 0F 0F 0F CD 21  F7F5....CD!
311F 31 F1 CD DA 03 12 13 C9  1F1CDDA...C9
3127 DD 6E 00 DD 66 01 7C CD  DD6E.DD66.7CCD
312F 18 31 7D CD 18 31 C9 11  .17DCD.1C9.
3137 2F 2D CD 15 00 3A F8 13  /-CD..:F8.
313F CB 47 C4 A5 01 3A A3 31  CBGC4u.:w1
3147 3C 32 A3 31 FE 17 28 01  <2w1FE.(.
314F C9 AF 32 A3 31 11 58 2D  C9j2w1.X-
3157 CD 15 00 CD 06 00 C5 CD  CD..CD..C5CD
315F 5D F0 C1 CD 1B 00 28 FB  ]F0C1CD..(£
3167 FE 66 C8 FE 64 CA 00 12  FE66C8FE64CA..
316F 18 F1 06 28 21 2F 2D 36  .F1.(!/-6
3177 20 23 10 FB C9 4E B7 B3   #.£C9Nom
317F 9A 9D 32 A4 20 98 32 9B  bres hex
3187 A1 2E 20 9C 32 20 A4 92  a. de se
318F 9F 96 92 A5 9D A4 20 B8  cteurs l
3197 A6 9A 9D 92 A4 3A 20 2D  ibres: -
319F 2D 2D 2D 0D 2B 2D 2B 2D  ---.+-+-
31A7 2D 2D 2D 2D 2D 2D 2D 2D  --------
31AF 2D 2D 2D 2D 2D 2D 2D 2D  --------
31B7 2B 2D 2D 2D 2D 2B 2D 2D  +----+--
31BF 2D 2D 2B 2D 2D 2D 2D 2B  --+----+
31C7 2D 2D 2D 2D 2B 0D 54 20  ----+.T
31CF 20 20 20 20 20 20 4E 4F        NO
31D7 4D 53 20 20 20 20 20 20  MS
31DF 20 20 B8 B7 B0 97 20 41    long A
31E7 9C 92 9A 20 41 3C 92 B3  deb Adem
31EF 20 41 A4 92 9F 20 0D A4   Asec .s
```

DUMP N° 2

```
1200 02 49 50 4C 50 52 4F 00  .IPLPRO.
1208 5A 45 4E 2D 58 50 2D 46  ZEN-XP-F
1210 44 0D 0D 0D 00 20 00 12  D.... ..
1218 00 12 00 00 00 FF 30 00  .....£0.
```

TORTUE GRAPHIQUE

**Grande lacune : dans le Sharpentier numéro 14, notre tortue graphique, amputée de son programme, avait bien du mal à vous dessiner les superbes graphiques promis.
Nous réparerons vite cet oubli, tellement énorme qu'il était passé au travers de nos vérifications.**

```
10 GOTO40
20 XXXXXXXXXX=DUMMY:RETURN
30 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
40 PROC"Initialise"
50 BEEP 3,9000,3
60 REPEAT
70 : Z=0:INPUT"@:";A$:BEEP 1,700,3
80 UNTIL FN"Interprete"(A$,0)=0
90 RUN 50
100 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
110 DEF FN"Etudie"(A$)
120 LOCAL I,N,C:N=0:C=0:NULL W$
130 N=FN"Words"(A$):E1=0
140 IF N=0 RESULT=1
150 IF W$(0)="FIN" RESULT=1
160 IF W$(0)="ENCORE" RESULT=1
170 E1=-1
180 FOR I=1 TO NBW
190 : A$=C$(I)
200 : IF RIGHT$(W$(0),1)="."
      A$=LEFT$(A$,LEN(W$(0))-1)+"."
210 : IF W$(0)=A$ C=I:C$=C$(I):I=NBW
220 NEXT
230 IF C=0 PROC"Err"(0) ELSE E1=0
240 RESULT=1
250 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
260 DEF FN"Interprete"(A$,P)
270 LOCAL I,N,C,J:N=0:C=0:NULL W$
280 N=FN"Words"(A$)
290 IF N=0 RESULT=1
300 IF W$(0)="FIN" RESULT=1
310 IF W$(0)="ENCORE" RESULT=1
320 E1=-1
330 FOR I=1 TO NBW
340 : A$=C$(I)
350 : IF RIGHT$(W$(0),1)="."
      A$=LEFT$(A$,LEN(W$(0))-1)+"."
360 : IF W$(0)=A$ C=I:C$=C$(I):I=NBW
370 NEXT
380 IF C=0 PROC"Err"(0):RESULT=1
390 E1=0
400 IF C<KW+1
      RESULT=EVAL("FN"+CHR$(34)+C$+CHR$(34))
410 Z=1
420 IF VAL(P$(C,0)) PROC"p" ELSE PROC"s"
430 BEEP 1,700,4:RESULT=1
440 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
450 DEF PROC"s"
460 LOCAL Q(1),Q(2),Q(3),Q(4),Q(5),C
470 REPEAT
480 : C=C+1:D=FN"Interprete"(C$(C),C)
490 : IF D=0 RUN 50
500 UNTIL C$(C)="FIN"
510 ENDPROC
520 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
530 DEF PROC"p"
540 LOCAL I
550 FOR I=1 TO VAL(P$(C,0))
560 : P(I)=FN"Eval"(I)
570 NEXT
580 FOR I=1 TO VAL(P$(C,0))
590 : Q(I)=EVAL(P$(C,I))
600 : PROC"Pass"(P$(C,I),P(I))
610 NEXT
620 PROC"s"
630 FOR I=1 TO VAL(P$(C,0))
640 : PROC"Pass"(P$(C,I),Q(I))
650 NEXT
660 ENDPROC
670 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
680 DEF FN"Eval"(I)
690 IF W$(I)="" RESULT=0
700 RESULT=EVAL(W$(I))
710 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
720 DEF FN"CHANGES"
730 MOVE 0,-480:HSET:A=0:X=0:Y=0
740 SI=0:CO=1:SCALE=1:CLS
750 PROC"Update":RESULT=1
760 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
770 DEF FN"LEVES":KS=0:RESULT=1
780 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
790 DEF FN"BAISSES":KS=1:RESULT=1
800 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
810 DEF FN"DROITE"
820 RESULT=FN"Direction"(-1)
830 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
840 DEF FN"GAUCHE"
850 RESULT=FN"Direction"(1)
860 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
870 DEF FN"Direction"(S)
880 A=(A+S*EVAL(W$(1)))MODULO 360
890 SI=SIN(RAD(A)):CO=COS(RAD(A))
900 PROC"Update":RESULT=1
910 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
920 DEF FN"LIGNE"
930 FL=EVAL(W$(1))
940 IF FL<=0 OR FL>16 FL=1
950 RESULT=1
960 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
970 DEF FN"AVANCES"
980 RESULT=FN"Trace"(1)
990 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1000 DEF FN"RECULES"
1010 RESULT=FN"Trace"(-1)
1020 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1030 DEF FN"Trace"(S)
1040 D=EVAL(W$(1)):X=X+S*D*CO*SCALE
1050 Y=Y+S*D*SI*SCALE
1060 X=X MAX -240 MIN 240
1070 Y=Y MAX -240 MIN 240
1080 IF KS=0 MOVE X,Y ELSE LINE %FL,X,Y
1090 PROC"Update":RESULT=1
1100 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1110 DEF FN"VAS-EN"
1120 X=EVAL(W$(1)):Y=EVAL(W$(2))
1130 X=X MAX -240 MIN 240
1140 Y=Y MAX -240 MIN 240
1150 MOVE X,Y:PROC"Update":RESULT=1
1160 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1170 DEF FN"COULEUR"
1180 LOCAL D:D=EVAL(W$(1))
1190 D=D MAX 0 MIN 3
1200 PCOLOR D:RESULT=1
1210 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1220 DEF FN"CENTRE"
1230 X=0:Y=0:A=0:SI=0:CO=1:MOVE 0,0
1240 PROC"Update":RESULT=1
1250 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1260 DEF FN"FINI"=0
1270 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1280 DEF PROC"Existe?"
1290 FOR J=KW+1 TO NBW
1300 IF W$(1)=C$(J) E=-1:J=NBW
1310 NEXT
1320 ENDPROC
1330 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1340 DEF FN"POUR":LOCAL J
1350 IF W$(1)="" PROC"Err"(1):RESULT=1
1360 E=0:IF NBW>KW PROC"Existe?"
1370 IF E PRINT W$(1);:PROC"Err"(2):RESULT=1
1380 J=NBW+1:C$(J)=W$(1):NBW=J
1390 P$(J,0)=STR$(N-2)
1400 IF N>2 PROC"Sauve parametres"
1410 REPEAT
1420 : J=J+1:INPUT"→→→:";C$(J)
1430 : D=FN"Etudie"(C$(J)):BEEP 1,500,2
1440 : IF E1=-1 J=J-1 ELSE NBW=NBW+1
1450 UNTIL C$(J)="FIN":BEEP 3,300,5
1460 NBW=J:RESULT=1
1470 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1480 DEF PROC"Sauve parametres"
1490 LOCAL I
1500 FOR I=1 TO N-2
1510 : P$(J,I)=W$(I+1)
1520 NEXT
1530 ENDPROC
1540 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1550 DEF FN"REPETES"
1560 LOCAL J
1570 IF Z=0 RESULT=1
1580 FOR J=1 TO EVAL(W$(1))-1
1590 : C=P+1
1600 : REPEAT
1610 :    D=FN"Interprete"(C$(C),C):C=C+1
1620 :    IF D=0 RUN 50
1630 : UNTIL C$(C)="ENCORE" OR C$(C)="FIN"
1640 NEXT J
1650 RESULT=1
1660 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1670 DEF FN"TAILLE"
1680 SCALE=SCALE*EVAL(W$(1))
1690 PROC"Update":RESULT=1
1700 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1710 DEF FN"ECHELLE"
1720 SCALE=EVAL(W$(1))
1730 PROC"Update":RESULT=1
1740 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1750 DEF FN"SOIT"
1760 PROC"Pass"(W$(1),EVAL(W$(2)))
1770 RESULT=1
1780 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1790 DEF PROC"Pass"(A$,DUMMY)
1800 LOCAL I,L
1810 A$=LEFT$(A$,10):L=LEN(A$)
1820 IF L<10 A$=A$+STRING$(10-L," ")
1830 FOR I=1 TO 10
1840 : POKE 30179+I,ASC(MID$(A$,I,1))
1850 NEXT
1860 GOSUB 20:ENDPROC
1870 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1880 DEF FN"Cherche"
1890 T=0
1900 IF W$(1)="" RESULT=0
1910 IF NBW=KW   RESULT=0
1920 FOR I=KW+1 TO NBW
1930 : IF W$(1)=C$(I) T=I:I=NBW
1940 NEXT
1950 RESULT=T
1960 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1970 DEF FN"PLISTES"
1980 MOVE -240,-240:HSET:MODE TN
1990 PLOT ON:D=FN"LISTES":PLOT OFF
2000 MODE GR:MOVE 240,-240:HSET
2010 PROC"Screen":PROC"Update"
2020 RESULT=1
2030 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
2040 DEF FN"LISTES"
2050 LOCAL T,I
2060 IF FN"Cherche">0 PROC"Ecris"
2070 RESULT=1
2080 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
2090 DEF PROC"Ecris"
2100 CLS:I=0
2110 PRINT"POUR ";W$(1):PROC"Parcmetres"
2120 REPEAT
2130 : T=T+1:I=I+1:PRINT I;TAB(4);C$(T)
2140 UNTIL C$(T)="FIN"
2150 ENDPROC
2160 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
2170 DEF PROC"Parametres"
2180 LOCAL I
2190 IF P$(T,0)="0" PRINT:ENDPROC
2200 PRINT"Parametres:";
2210 FOR I=1 TO VAL(P$(T,0))
2220 : PRINT " "+P$(T,I);
2230 NEXT
2240 PRINT
2250 ENDPROC
2260 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
2270 DEF FN"OUBLIES"
2280 LOCAL T,T1,I,J:T1=0
2290 IF FN"Cherche"=0 RESULT=1
2300 T1=T:P$(T,0)=""
2310 REPEAT:T=T+1:UNTIL C$(T)="FIN"
2320 FOR I=0 TO NBW-T-1
2330 : C$(T1+I)=C$(T+I+1)
2340 : FOR J=0 TO 5
2350 :    P$(T1+I,J)=P$(T+I+1,J)
2360 : NEXT
2370 NEXT
2380 NBW=NBW+T1-T-1
2390 RESULT=1
2400 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
2410 DEF FN"SAUVES"
2420 LOCAL I,J
2430 PRINT CHR$($7F)+" Record.Play"
2440 WOPEN W$(1):PRINT/T NBW
2450 FOR I=KW+1 TO NBW
2460 : PRINT/T C$(I)
2470 : FOR J=0 TO 5
2480 :    PRINT/T P$(I,J)
2490 : NEXT
2500 NEXT
2510 CLOSE
2520 RESULT=1
2530 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
2540 DEF FN"CHARGES"
2550 LOCAL I,J
2560 PRINT CHR$($7F)+" Play"
2570 ROPEN W$(1):INPUT/T NBW
2580 FOR I=KW+1 TO NBW
2590 : INPUT/T C$(I)
2600 : FOR J=0 TO 5
2610 :    INPUT/T P$(I,J)
2620 : NEXT
2630 NEXT
2640 CLOSE
2650 RESULT=1
2660 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
2670 DEF FN"Words"(A$)
2680 IF LEN(A$)=0 RESULT=0
2690 WHILE LEFT$(A$,1)=" " DO
2700 : A$=IN$(A$,0,2,"")
2710 WEND
2720 N=N+1:IF N=8 RESULT=7
2730 PROC"Espace"
2740 IF LEFT$(A$,1)="[" PROC"Crochet"
2750 W$(N-1)=B$:IF I=0 RESULT=N
2760 RESULT=FN"Words"(RIGHT$(A$,LEN(A$)-I))
2770 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
2780 DEF PROC"Crochet"
2790 I=INSTR(A$,"]",1)
2800 IF I=0 B$=IN$(A$,0,2,""):ENDPROC
2810 B$=MID$(A$,2,I-2):ENDPROC
2820 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
2830 DEF PROC"Espace"
2840 I=INSTR(A$," ",1)
2850 IF I=0 B$=A$ ELSE B$=LEFT$(A$,I-1)
2860 ENDPROC
2870 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
2880 DEF PROC"Update"
2890 LOCAL S,XO,YO
2900 XO=HPOS:YO=VPOS
2910 CONSOLE 0,11,30,10:COLOR,,,2:CLS
2920 GRAPH 69,10
2930 DRAW 1,69+10*CO,10-10*SI,4
2940 CONSOLE 0,11,3,6:COLOR ,,3,6
2950 S=INT(SCALE*100)/100:PRINT"■■";
2960 PRINT"■■";INT(X):PRINT"■";INT(Y)
2970 PRINT"■";S:PRINT"■";A
2980 CONSOLE 11,14,0,40:CURSOR XO,YO
2990 COLOR ,,1,5:ENDPROC
3000 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
3010 DEF PROC"Err"(E)
3020 PRINT ER$(E):BEEP 10,450,3
3030 ENDPROC
3040 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
3050 DEF PROC"Initialise"
3060 MODE GR:MOVE 240,-240:HSET
3070 PCOLOR 0:MODE 0:CLR
```

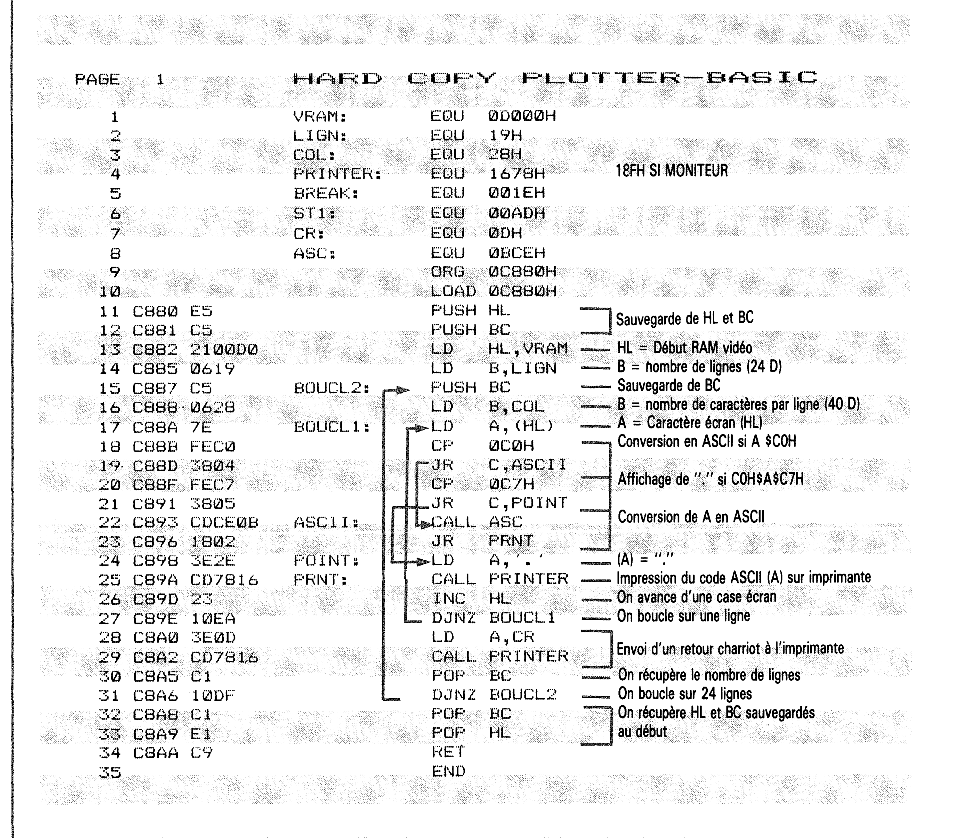
```
3080 KS=1:FL=1:E1=-1:KW=25:NBW=KW
3090 A=0:SI=0:CO=1:X=0:Y=0:SCALE=1
3100 DIM C$(100),P$(100,5)
3110 DIM W$(6),P(5),Q(5),ER$(2)
3120 FOR I=1 TO KW:READ C$(I):NEXT
3130 FOR I=0 TO 2:READ ER$(I):NEXT
3140 PROC"Screen":PROC"Update"
3150 NOIR=0:BLEU=1:VERT=2:ROUGE=3
3160 DEF KEY(9)="CLS:RUN 50"+CHR$(13)
3170 DEF KEY(10)="MODETN:SKIP-10"+CHR$(13)
3180 ENDPROC
3190 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
3200 DATA CHANGES,LEVES,BAISSES,DROITE
3210 DATA GAUCHE,AVANCES,VAS-EN,COULEUR
3220 DATA FINI,POUR,REPETES,TAILLE,SOIT
3230 DATA SAUVES,CHARGES,RECULES,CENTRE
3240 DATA LISTES,PLISTES,ECHELLE
3250 DATA LIGNE,EDITES,SI,SINON,OUBLIES
3260 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
3270 DATA "Pas encore defini !"
3280 DATA "Pour quoi ?"
3290 DATA " deja definie !"
3300 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
3310 DEF PROC"Screen"
3320 CONSOLE:CLS
3330 CONSOLE 0,11,9,21:COLOR ,,0,4:CLS
3340 PRINT "██████T O R T U E"
3350 PRINT "█████~~~~~~~~~~~"
3360 PRINT "████G R A P H I Q U E"
3370 PRINT "██~~~~~~~~~~~~~~~~~"
3380 CONSOLE 0,11,0,10:COLOR ,,3,6:CLS
3390 PRINT"██ X=████Y=████S=████A="
3400 CONSOLE 11,14,0,40:COLOR ,,1,5:CLS
3410 ENDPROC
3420 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
3430 DEF PROC"Editeur"
3440 CONSOLE 0,25,0,19:COLOR ,,6,2:CLS
3450 T1=T:W$(1)=C$(T):PROC"Ecris"
3460 PRINT"█E ffface  ligne"
3470 PRINT"M odifie ligne"
3480 PRINT"I nsere  ligne"
3490 PRINT"F in editeur"
3500 CONSOLE 0,25,19,21:COLOR ,,0,7:CLS
3510 ENDPROC
3520 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
3530 DEF FN"EDITES"
3540 LOCAL T,I,T1,R$
3550 IF FN"Cherche"=0 RESULT=-1
3560 T1=T
3570 REPEAT
3580 : T=T1:PROC"Editeur"
3590 : REPEAT
3600 :    INKEY R$
3610 : UNTIL INSTR("EMIF",R$,1)>0
3620 : D=EVAL("FN"+CHR$(34)+R$+CHR$(34))
3630 UNTIL D=0
3640 RESULT=1
3650 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
3660 DEF FN"F"
3670 BEEP 1:PROC"Screen":PROC"Update"
3680 RESULT=0
3690 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
3700 DEF PROC"Numero"(L)
3710 REPEAT
3720 INPUT"ligne n°";N:N=ABS(INT(N))
3730 UNTIL N>L AND N<I
3740 ENDPROC
3750 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
3760 DEF PROC"Entre"
3770 REPEAT
3780 : INPUT"++:";A$:D=FN"Etudie"(A$)
3790 UNTIL E1=0
3800 ENDPROC
3810 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
3820 DEF FN"M"
3830 PRINT"Modification de la":BEEP 1
3840 PROC"Numero"(-1):PROC"Entre"
3850 C$(N+T1)=A$:RESULT=1
3860 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
3870 DEF FN"E"
3880 LOCAL J,K
3890 PRINT"Effacement de la":BEEP 1
3900 PROC"Numero"(0)
3910 FOR J=T1+N+1 TO NBW
3920 : C$(J-1)=C$(J)
3930 : FOR K=0 TO 5
3940 :    P$(J-1,K)=P$(J,K)
3950 : NEXT K
3960 NEXT J
3970 NBW=NBW-1:I=I-1
3980 RESULT=1
3990 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
4000 DEF FN"I"
4010 LOCAL J,K
4020 PRINT"Insertion apres la":BEEP 1
4030 PROC"Numero"(-1)
4040 FOR J=NBW TO T1+N+1 STEP -1
4050 : C$(J+1)=C$(J)
4060 : FOR K=0 TO 5
4070 :    P$(J+1,K)=P$(J,K)
4080 : NEXT K
4090 NEXT J
4100 NBW=NBW+1:I=I+1:PROC"Entre"
4110 C$(T1+N+1)=A$:RESULT=1
4120 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
4130 DEF FN"SI"
4140 LOCAL K,I,J,A$
4150 K=EVAL(W$(1)):D=1
4160 IF K<>0 PROC"Then" ELSE PROC"Else"
4170 IF A$<>"" D=FN"Interprete"(A$,P)
4180 IF D=0 RUN 50
4190 RESULT=1
4200 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
4210 DEF PROC"Then"
4220 I=INSTR(C$(P),W$(1),1)
4230 A$=IN$(C$(P),0,I+LEN(W$(1))+1,"")
4240 ENDPROC
4250 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
4260 DEF PROC"Else"
4270 I=INSTR(C$(P),"SINON",1):A$=""
4280 IF I>0 A$=IN$(C$(P),0,I+6,"")
4290 ENDPROC
4300 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
4310 DEF FN"SINON"=1
4320 '~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```


HARD COPY

Dans la littérature informatique, il est souvent bien difficile de défendre la langue de Bossuet face aux assauts du jargon anglo-saxon. Là où nous disons RECOPIE D'ECRAN SUR IMPRIMANTE, les Outre-Manchois interjectent HARD COPY ; soit une économie de 18 lettres et 6 syllabes, évidemment Bossuet n'utilisait pas encore de traitement de texte...

Les routines que vous propose M. PANETTO vont donc vous permettre de recopier, à tout moment, l'intégralité des textes affichés à l'écran sur votre imprimante. La routine 1 permet d'utiliser la table traçante SHARP sous moniteur ou sous BASIC standard (S ou K.BASIC). Si vous possédez un BASIC CENTRO, cette routine fonctionnera également pour toute imprimante au standard CENTRONICS. La routine 2 fonctionnera avec les BASIC standards et une imprimante CENTRO. Outre l'intérêt pratique de ces 2 utilitaires, l'étude de la routine 1 et de ses commentaires permettra aux apprentis sorciers du L.M. de découvrir plusieurs techniques nouvelles de programmation (sauts relatifs, boucles, sauvegarde de registres) que nous aborderons dans nos futures leçons.

S.B.



```
PAGE   1              HARD  COPY  PLOTTER-BASIC

                 VRAM:      EQU  0D000H
                 LIGN:      EQU  19H
                 COL:       EQU  28H
                 PRINTER:   EQU  1678H       18FH SI MONITEUR
                 BREAK:     EQU  001EH
                 ST1:       EQU  00ADH
                 CR:        EQU  0DH
                 ASC:       EQU  0BCEH
                            ORG  0C880H
                            LOAD 0C880H
C880 E5                     PUSH HL        ┐ Sauvegarde de HL et BC
C881 C5                     PUSH BC        ┘
C882 2100D0                 LD   HL,VRAM   — HL = Début RAM vidéo
C885 0619                   LD   B,LIGN    — B = nombre de lignes (24 D)
C887 C5          BOUCL2:    PUSH BC        — Sauvegarde de BC
C888 0628                   LD   B,COL     — B = nombre de caractères par ligne (40 D)
C88A 7E          BOUCL1:    LD   A,(HL)    — A = Caractère écran (HL)
C88B FEC0                   CP   0C0H      — Conversion en ASCII si A $C0H
C88D 3804                   JR   C,ASCII
C88F FEC7                   CP   0C7H      — Affichage de "." si C0H$A$C7H
C891 3805                   JR   C,POINT
C893 CDCE0B      ASCII:     CALL ASC       — Conversion de A en ASCII
C896 1802                   JR   PRNT
C898 3E2E        POINT:     LD   A,'.'     — (A) = "."
C89A CD7816      PRNT:      CALL PRINTER   — Impression du code ASCII (A) sur imprimante
C89D 23                     INC  HL        — On avance d'une case écran
C89E 10EA                   DJNZ BOUCL1    — On boucle sur une ligne
C8A0 3E0D                   LD   A,CR      ┐ Envoi d'un retour charriot à l'imprimante
C8A2 CD7816                 CALL PRINTER   ┘
C8A5 C1                     POP  BC        — On récupère le nombre de lignes
C8A6 10DF                   DJNZ BOUCL2    — On boucle sur 24 lignes
C8A8 C1                     POP  BC        ┐ On récupère HL et BC sauvegardés
C8A9 E1                     POP  HL        ┘ au début
C8AA C9                     RET
                            END
```

```
PAGE   1          HARD COPY TEXTE          MZ 700
                  VRAM:     EQU  0D000H
                  LIGN:     EQU  19H
                  COL:      EQU  28H
                  BREAK:    EQU  001EH
                  ST1:      EQU  00ADH
                  CR:       EQU  0DH
                  ASC:      EQU  0BCEH
                            ORG  0C880H
                            LOAD 0C880H
C880 E5                     PUSH HL
C881 C5                     PUSH BC
C882 2100D0                 LD   HL,VRAM
C885 0619                   LD   B,LIGN
C887 C5           BOUCL2:   PUSH BC
C888 0628                   LD   B,COL
C88A 7E           BOUCL1:   LD   A,(HL)
C88B FEC0                   CP   0C0H          ;CTRL
C88D 300D                   JR   NC,POINT
C88F FE80                   CP   80H           ;DISP /-1
C891 3804                   JR   C,ASCII
C893 FE9B                   CP   9BH           ;DISP "+1
C895 3807                   JR   C,PRNT
C897 CDCE0B       ASCII:    CALL ASC
C89A 1802                   JR   PRNT
C89C 3E2E         POINT:    LD   A,'.'
C89E CDAFC8       PRNT:     CALL PRINTER
C8A1 23                     INC  HL
C8A2 10E6                   DJNZ BOUCL1
C8A4 3E0D                   LD   A,CR
C8A6 CDAFC8                 CALL PRINTER
C8A9 C1                     POP  BC
C8AA 10DB                   DJNZ BOUCL2
C8AC C1                     POP  BC
C8AD E1                     POP  HL
C8AE C9                     RET
C8AF C5           PRINTER:  PUSH BC
C8B0 0E00                   LD   C,00H
C8B2 47                     LD   B,A
C8B3 CDC7C8                 CALL RDA
C8B6 78                     LD   A,B
C8B7 D3FF                   OUT  (0FFH),A
C8B9 AF                     XOR  A
C8BA D3FE                   OUT  (0FEH),A
C8BC 0E01                   LD   C,01H
C8BE CDC7C8                 CALL RDA
C8C1 3E80                   LD   A,80H
C8C3 D3FE                   OUT  (0FEH),A
C8C5 C1                     POP  BC
C8C6 C9                     RET
C8C7 DBFE         RDA:      IN   A,(0FEH)
C8C9 E601                   AND  01H
C8CB B9                     CP   C
C8CC C8                     RET  Z
C8CD CD1E00                 CALL BREAK
C8D0 20F5                   JR   NZ,RDA
C8D2 C3AD00                 JP   ST1
                            END
```

ASTUCES

ASTUCES



UN USR bien placé dans votre programme vous permettra d'effectuer un déplacement de la totalité de votre écran vers le haut ou le bas (SCROLL).
Sous S. BASIC : USR ($655) vers le haut.
USR ($672) vers le bas.
Sous DISK BASIC URS ($64B) vers le haut.
($672) vers le bas.



Comment simuler la fonction LINE SUR MZ avec les pavés graphiques, il vous suffit de taper le sous-programme suivant :

```
60000 D9=SQR(X9-X8) 2+(Y9-Y8) 2)
60010 R9=(X9-X8)/D9
60020 59=(Y9-Y8) D9
60030 FORI=OTOD+.5:SETX8,Y8
60040 X8=X8+R9:Y8=Y8+S9:NEXT
60050 RETURN
```

X8 et Y8 sont les coordonnées du premier point, X9 et Y9 les coordonnées du second point. Attention, ces valeurs ne doivent pas être égales.

EXEMPLE :

X8=0:Y8=0:X9=79:Y9=48:GOSUB 60000:END

M. Szczepanski



Un moyen simple de sélectionner définitivement les minuscules et majuscules sur MZ : Il suffit de taper simultanément SHIFT et ALPHA et de recommencer la même opération pour revenir au mode majuscule.

N. Le Nours.

LM SUR MZ

Leçon N° 5 où nous allons aborder la fonction essentielle d'un microprocesseur : Le calcul arithmétique et étudier un utilitaire permettant de comprendre aisément l'action des instructions de calculs sur les différents registres du Z80.

Z80 ET ARITHMETIQUE

Il faut bien le reconnaître, le Z80, comme tous les microprocesseurs 8 bits, a tout juste le niveau d'un élève de cours élémentaire puisque ses connaissances en calcul se limitent à quelques opérations de base : Addition, soustraction, comparaison et manipulation d'octets. C'est à partir de ces éléments primaires que s'effectuent tous les calculs mathématiques présents dans les langages de haut niveau (BASIC, PASCAL, FORTH, etc). Je vous laisse deviner la joie ineffable que procure, par exemple, pour le programmeur, l'extraction d'une racine carrée en virgule flottante à partir de ces bien maigres possibilités. Heureusement, nous l'avons déjà constaté, le Z80 travaille très vite ; ceci compensant cela, votre racine carrée n'aura besoin que de quelques millisecondes pour être calculée.

L'UTILITAIRE « REGISTRES »

Avant d'entrer plus avant dans le détail de ces instructions, je vous propose de taper et d'étudier le programme REGISTRES ci joint. Cet utilitaire vous permettra d'entrer quelques codes L.M. en Hexa constituant le mini-programme à étudier et de sauter à un ensemble de routines L.M. puis BASIC qui initialiseront les registres du Z80, exécuteront votre programme puis afficheront les registres du microprocesseur. Vous pourrez ainsi étudier à loisirs l'action des différentes instructions de votre routine sur ces registres.

ETUDE DU PROGRAMME LIGNE A LIGNE

- 5- Met à zéro les adresses mémoire comprises entre $BFCO et $CO40.
- 7-40 - Implante en mémoire, de $BFD7 à $BFFF la première partie de la routine L.M. dont le rôle est de mettre à zèro tous les registres généraux du Z80. Cette routine est chargée grâce aux DATA's des lignes 1010 à 1170.
- 50-80 - Implante la seconde partie de la routine L.M., de $C020 à $C058 (DATA's des lignes 1230 à 1450) qui se charge de transférer le contenu des registres du Z80 dans une zone mémoire comprise entre $B800 et $B817.
- 110-170 - Routine BASIC d'entrée du mini-programme à étudier, ce programme s'implante de $C000 à $C01F, soit 32 octets maximum. L'entrée se fait en HEXA, code par code, après affichage par le MZ de l'adresse correspondante. Après l'entrée du code, la valeur pokée est affichée sur la même ligne pour vérification.

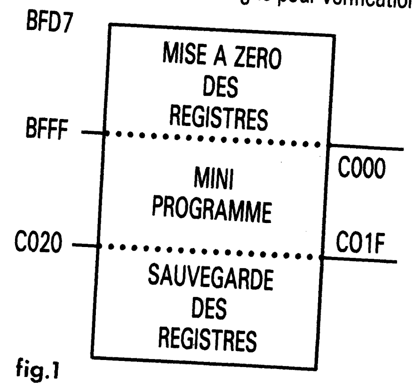


fig.1

En cas d'erreur de saisie d'un code, il suffit de taper SHIFT « * » pour effectuer une nouvelle saisie à la même adresse. Le programme teste également la longueur de l'octet saisi qui ne doit pas dépasser 2 caractères. Lorsque le dernier octet de votre routine est entré, vous avez donc un programme L.M. prêt à être exécuté, composé de 3 parties. (Fig. 1) La première partie met à zéro les registres du microprocesseur ; la seconde partie exécute votre routine. Si celle-ci est inférieure à 32 octets, les octets non utilisés seront à zéro ; le Z80 les interprétera comme une instruction appelée NOP (NO OPERATION) qui, comme son nom l'indique a pour effet de ne rien faire exécuter au Z80 sinon de passer à l'instruction suivante. Il sautera ainsi, par NOP's successifs à la routine de sauvegarde en mémoire de ses registres.

- 500 - C'est la ligne d'appel du programme L.M. Pour la déclencher, il vous suffit, en fin de saisie, de taper « @ ». Celui-ci se terminant par un RET, rendra la main au programme BASIC.
- 510-630 - Le programme BASIC continue et affiche à l'écran, en clair, le contenu des différents registres en allant chercher leurs valeurs aux adresses où elles ont été sauvegardées.
- 900-915 - Sous-programme qui affiche en HEXA le contenu d'une adresse mémoire.
- 920-950 - Sous-programme qui teste chaque bit du registre F et affiche son contenu à côté du flag correspondant.
- 1000-1170 - DATA's de la routine de remise à zéro des registres. Elle aurait pu être plus courte mais moins explicite. Comme aucune instruction ne permet de mettre à zéro le registre F, il a fallu utiliser une astuce (lignes 1060-1070) qui consiste à mettre le contenu de HL, soit $0000 dans la pile (PUSH HL) et à rappeler ce contenu pour le placer dans A et F (POP AF). Cela se reproduit pour les registres « PRIME » en lignes 1090 et 1100. Pour éviter tout problème, le pointeur de pile qui contient l'adresse de retour au BASIC a été sauvegardé en $BA00 (ligne 1160) ; il sera rappelé juste avant la fin de la routine (ligne 1440). Un nouveau pointeur de pile est créé, nous l'utiliserons pour étudier les effets de certaines instructions sur la pile, il se situe en $BB00 (ligne 1170).
- 1230-1450 - Routine de sauvegarde des registres du Z80 après exécution du mini-programme saisi. A noter une nouvelle utilisation de la pile pour transférer les registres A et F dans HL et sauvegarder ainsi leurs contenus en mémoire (lignes 1260-1270/1350-1360).

Ces explications ont peut-être été un peu longues et fastidieuses, mais elles étaient indispensables pour vous permettre de bien comprendre comment fabriquer et utiliser une routine en L.M. à partir d'un programme BASIC.

PREMIERES APPLICATIONS

Comme pour toutes les applications suivantes et futures, il vous suffira de taper les codes HEXA correspondants aux instructions du mini-

programme puis de taper « @ » pour l'exécuter et obtenir l'affichage des registres.
Imaginons que nous voulions placer dans A la somme du contenu des registres B et C. Commençons par placer 2 valeurs quelconques dans B et C : LD B, $18 – LD C, $2C. Ensuite, le premier terme de l'addition est placé dans A : LD A,B puis le second terme ajouté au premier : ADD A,C. cela nous donne le programme suivant :

```
06 18   LD B,$18
0E 2C   LD C,$2C
78      LD A,B
81      ADD A,C
```

Le registre A contient maintenant la somme de B+C. L'instruction fonctionne en 8 ou 16 bits. En 8 bits, elle additionne le contenu de n'importe quel registre général ou valeur numérique au registre A. En 16 bits, les registres supports peuvent être HL, IX ou IY. Essayez par exemple :

```
21 00 10   LD HL,$1000
01 50 18   LD BC,$1850
09         ADD HL,BC
```

HL contiendra $2850, soit HL+BC.

L'instruction ADD sur 8 bits peut s'utiliser avec un registre dit d'INDEX qui peut être (HL), (IX+d)(IY+d). Nous reviendrons ultérieurement sur les registres d'index IX et IY ; étudions (HL).

Tapez :

```
21 20 C0   LD HL,$C020
3E 20      LD A,$20
86         ADD A,(HL)
```

A contiendra $42 soit son propre contenu ajouté au contenu de l'adresse POINTEE par HL. HL contient la valeur $C020 qui est le numéro d'une adresse. Cette adresse contient la valeur $22(c'est la première adresse du programme de sauvegarde des registres) ; c'est cette valeur, contenue dans l'adresse pointée par HL qui sera ajoutée à A. Cette notion de POINTEUR que nous avons déjà abordée dans notre précédente leçon, se retrouve dès qu'une valeur ou un registre 16 bits sont inscrits entre () dans une instruction .
Pour compléter cette notion fondamentale, imaginons que nous voulions mettre dans A la somme du contenu des adresses $BFD7 à $BFBD, soit 7 adresses successives ; il nous suffit, dans ce cas , d'entrer :

```
21 D7 BF  LD HL,$BFD7
7E        LD A, (HL)
23 INC HL              ┌─► 23 INC HL
86 ADD A, (HL)         │   86 ADD A, (HL)
23 INC HL              │   23 INC HL
86 ADD A, (HL)         │   86 ADD A, (HL)
23 INC HL              │   23 INC HL
86 ADD A, (HL) ────────┘   86 ADD A, (HL)
```

Nous avons alors dans A le contenu des 7 premières adresses du sous-programme de mise à zéro des registres (ouf !) soit $21+0+0+$11+0+0+$01
Heureusement, on arrive à faire beaucoup plus court pour obtenir le même résultat, mais ce n'est pas notre préoccupation pour l'instant.
L'instruction ADD a un avantage et un inconvénient. Avantage : Elle utilise pratiquement tous les registres du Z80, elle est donc très souple d'emploi. Inconvénient : Le programmeur n'a aucun moyen de savoir si le résultat de l'opération dépasse les capacités d'un registre 8 bits (0 à 255) ou 16 bits (0 à 65535). Pour pallier à cette lacune, les concepteurs du Z80 ont prévu un second jeu d'instructions appelé ADC dont le fonctionnement est identique à ADD et qui n'en diffère que par le C qui signifie CARRY (retenue). En cas de dépassement de capacité (résultat sup. à 255 ou 65535), le bit C du registre F se positionne à 1 ; il suffira ensuite de tester ce bit pour être renseigné sur la nature du résultat de l'opération. Par extension, on pourra effectuer des opérations sur 9 ou 17 bits mais ne nous égarons pas...
La pratique valant toujours mieux qu'un long discours, n'hésitez pas à faire de nombreux essais à l'aide de REGISTRES autour de ces 2 instructions et à analyser les résultat de ces essais. Pourquoi ensuite, ne pas vous aventurer à utiliser les instructions que nous avons déjà abordées dans nos précédentes leçons et à les intégrer dans vos programmes : LD (LOAD avec ou sans notion d'index), CP (comparaison du contenu de A avec celui des autres registres), DEC (décrémenter = – 1), INC (son inverse), JP (Saut conditionnel ou non à une adresse). Vous pouvez également tester les 2 instructions inverses à ADD et ADC qui sont SUB(Soustraction sans retenue) et SBC (Devinez...).

Un dernier conseil : N'entreprenez aucun essai avant d'avoir enregistré le programme REGISTRES car une erreur de saisie pourrait provoquer un plantage de la routine entrée. Dans la majorité des cas, un CTRL-RESET suffira pour récupérer BASIC et programme.
BON COURAGE...

SYLVAIN BIZOIRRE

```
3 L$=SPACE$(39)
5 FORI=$BFC0TO$C040:POKEI,0:NEXT
7 RESTORE 1000
10 FORI=$BFD7TO$BFFF
20 READ A
30 POKEI,A
40 NEXTI
50 FORI=$C020TO$C05B
60 READ A
70 POKEI,A
80 NEXTI
100 CLS
110 AD=$C000
120 PRINTHEX$(AD);:INPUT" ";T$
125 IFLEN(T$)>2THENBEEP:PRINT"■";L$;"■■":GOTO120
130 IFT$="*"THENBEEP:PRINT"■■";L$:PRINTL$;"■■"
    :POKEAD,0:AD=AD-1:GOTO120
133 T=VAL("$"+T$)
135 IFT$="@"THEN500
140 TR$=HEX$(T):IFLEN(TR$)=1THENTR$="0"+TR$
145 POKEAD,T
150 PRINT"■";TAB(10);TR$
160 AD=AD+1:IFAD=$C01FTHENPRINT"Attention
    derniere adresse valide."
170 GOTO120
500 USR($BFD7)
510 PRINT" A = ";:H=$B805:GOSUB900:PRINT
520 PRINT" F = ";:H=$B804:GOSUB900:GOSUB920:PRINT
530 PRINT"BC = ";:H=$B809:GOSUB900:H=H-1:GOSUB900:PRINT
540 PRINT"DE = ";:H=$B80B:GOSUB900:H=H-1:GOSUB900:PRINT
550 PRINT"HL = ";:H=$B801:GOSUB900:H=H-1:GOSUB900:PRINT
560 PRINT"IX = ";:H=$B80D:GOSUB900:H=H-1:GOSUB900:PRINT
570 PRINT"IY = ";:H=$B80F:GOSUB900:H=H-1:GOSUB900:PRINT
580 PRINT" A'= ";:H=$B811:GOSUB900:PRINT
590 PRINT" F'= ";:H=$B810:GOSUB900:GOSUB920:PRINT
600 PRINT"BC'= ";:H=$B815:GOSUB900:H=H-1:GOSUB900:PRINT
610 PRINT"DE'= ";:H=$B817:GOSUB900:H=H-1:GOSUB900:PRINT
620 PRINT"HL'= ";:H=$B813:GOSUB900:H=H-1:GOSUB900:PRINT
630 PRINT"SP = ";:H=$B803:GOSUB900:H=H-1:GOSUB900:PRINT
890 END
900 A$=HEX$(PEEK(H))
905 IFLEN(A$)=1THENA$="0"+A$
910 PRINTA$;
915 RETURN
920 C=256:D=PEEK(H):RESTORE950:PRINT"     ";
925 C=C/2:E=0:IFC<1THENRETURN
930 IFD>=CTHENE=1:D=D-C
935 READF$:IFF$="X"THEN925
940 PRINTF$;"=";STR$(E);" ";
945 GOTO925
950 DATA"S","Z","X","H","X","P/V","N","C"
1000 ' === MISE A ZERO DES REGISTRES ===
1010 DATA $21,$00,$00        :'LD HL,0000H
1020 DATA $11,$00,$00        :'LD DE,0000H
1030 DATA $01,$00,$00        :'LD BC,0000H
1040 DATA $DD,$21,$00,$00    :'LD IX,0000H
1050 DATA $FD,$21,$00;$00    :'LD IY,0000H
1060 DATA $E5                :'PUSH HL
1070 DATA $F1                :'POP  AF
1080 DATA $08                :'EX   AF,AF'
1090 DATA $E5                :'PUSH HL
1100 DATA $F1                :'POP  AF
1105 DATA $08                :'EX   AF,AF'
1110 DATA $D9                :'EXX
1120 DATA $21,$00,$00        :'LD HL,0000H
1130 DATA $11,$00,$00        :'LD DE,0000H
1140 DATA $01,$00,$00        :'LD BC,0000H
1150 DATA $D9                :'EXX
1160 DATA $ED,$73,$00,$BA    :'LD(BA00),SP
1170 DATA $31,$00,$BB        :'LD SP,BB00H
1200 '
1210 '=== SAUVEGARDE DES REGISTRES ===
1220 '
1230 DATA $22,$00,$B8        :'LD(B800),HL
1240 DATA $ED,$73,02,$B8     :'LD(B802),SP
1260 DATA $F5                :'PUSH AF
1270 DATA $E1                :'POP HL
1280 DATA $22,$04,$B8        :'LD(B804),HL
1290 DATA $ED,$73,$06,$B8    :'LD(B806),SP
1300 DATA $ED,$43,$08,$B8    :'LD(B808),BC
1310 DATA $ED,$53,$0A,$B8    :'LD(B80A),DE
1320 DATA $DD,$22,$0C,$B8    :'LD(B80C),IX
1330 DATA $FD,$22,$0E,$B8    :'LD(B80E),IY
1340 DATA $08                :'EX AF,AF'
1350 DATA $F5                :'PUSH AF
1360 DATA $E1                :'POP HL
1370 DATA $22,$10,$B8        :'LD(B810),HL
1380 DATA $08                :'EX AF,AF'
1390 DATA $D9                :'EXX
1400 DATA $22,$12,$B8        :'LD(B812),HL
1410 DATA $ED,$43,$14,$B8    :'LD(B814),BC
1420 DATA $ED,$53,$16,$B8    :'LD(B816),DE
1430 DATA $D9                :'EXX
1440 DATA $ED,$7B,$00,$BA:'LD SP,(BA00)
1450 DATA $C9                :'RET
```

PITMAN

PITMAN est un jeu inspiré d'une revue japonaise et adapté au S.BASIC. Il fonctionne donc sur MZ 700 et MZ 800 en mode 700 avec BASIC 700.

Ce jeu est passionnant à plus d'un titre. D'abord parce qu'il constitue une suite inépuisable de casse-têtes à résoudre, ensuite parce qu'il fait participer activement le joueur à la création d'autres problèmes.
Le programme comporte 50 écrans graphiques différents, à difficulté croissante, comprenant chacun 1 homme (Pitman) et 1 ou plusieurs diamants. Le but de l'opération est de faire accèder Pitman à ce(s) diamant(s). Pour s'aider, Pitman peut utiliser des échelles et des pavés hexagonaux qu'il pousse à volonté. 2 types de carrés sont également à sa disposition : les carrés bleus, indestructibles mais bien utiles pour se déplacer et les carrés rouges que Pitman peut détruire pour se déplacer. Comme tout jeu de qualité, les règles sont très simples mais vous vous apercevrez que l'objectif à atteindre l'est beaucoup moins. Cela peut même durer de longues heures qui se prolongent tard... tard dans la nuit (expérience vécue).
L'intérêt de Pitman ne s'arrête pas là. Il permet, en effet, de créer votre propre fichier de tableaux grâce à un éditeur graphique simple et pratique.

Entrons dans les détails :
Tapez le programme, vérifiez-le très soigneusement car certaines instructions sont complexes et les lignes de DATA's ne souffrent aucune erreur. Sauvegardez votre programme et tapez ensuite le fatidique RUN.
Après quelques secondes d'initialisation, le titre s'affiche avec le choix entre le jeu et le moniteur graphique. Si vous désirez jouer et ne pas commençer par le premier tableau, vous pouvez taper un chiffre entre 1 et 9 qui vous affichera le tableau du chiffre tapé ★ 5. En cours de jeu, il vous suffit de déplacer Pitman avec les touches curseur. Vous pouvez, à tout moment, interrompre le jeu en tapant « S » (tableau suivant), « P » (tableau précédent) ou « M » (réaffichage du même tableau). Attention cependant, le fait de changer de tableau vous fait perdre 1 Pitman, et vous n'en avez que 5. Une dernière commande : « E » vous permet de passer au moniteur intégré dont les commandes sont les suivantes :
A : Afficher la liste des commandes disponibles dans le moniteur.
J : Retourner au jeu.
L : Lire un fichier de 50 écrans.
E : Effacer 1 écran.
D : Activer l'éditeur d'écran pour créer de nouveaux tableaux.
R : Enregistrer un fichier de 50 écrans.
T : Retour au début du jeu avec initialisation des écrans d'origine.
Quelques commentaires sur les commandes de ce moniteur. La commande R enregistre tous les écrans en cours au moment de la commande y compris ceux éventuellement effacés, modifiés ou créés. L'éditeur d'écran permet de modifier un écran existant ou de créer un écran précédemment effacé. Le curseur graphique se déplace à l'aide des touches curseur.
Pour modifier une case, taper sur les touches 1 à 7 :
1 Pour effacer la case.
2 Pour afficher un morceau d'échelle.
3 Pour afficher un diamant.
4 Pour afficher un carré rouge.
5 Pour afficher un pavé hexagonal.
6 Pour afficher un carré bleu.
7 Pour afficher 1 Pitman.
M Pour retourner au moniteur.
P Pour afficher le tableau précédent.
S Pour afficher le tableau suivant.
Voilà, il ne vous reste plus qu'à vous passionner pour ce jeu comme nous l'avons fait nous-même, à créer de nouvelles difficultés – Attention, ce n'est pas facile –, à conçevoir vos propres jeux graphiques en vous aidant des idées apportées par PITMAN... et à nous en faire profiter.

S. B.

```
10 DEFKEY(1)="GOTO100"+CHR$(13)
20 B$="■J'initialise, patience, c'est fini.":RESTORE2590:X=$BE00
30 READA$:B=B+1:IFA$="F"THENPRINT:GOTO60
40 PRINTMID$(B$,B,1);:FORI=1TO128:V=VAL(MID$(A$,I,1)):POKEX,V:X=X+1:Y=Y+V
50 NEXT:GOTO30
60 IF(PEEK($CF2E)<>1)+(Y<>7837)THENPRINT"Erreur dans les DATA's.":END
70 ' ┌────────┐
80 ' | Titre  |
90 ' └────────┘
100 COLOR,,7,0:CONSOLE:CLS:RESTORE:GOSUB2250
110 GETA$:IFA$=""THEN110
120 IFA$="M"THEN1040
130 IFVAL(A$)>0THENL=VAL(A$)*5-1:GOTO160
140 IFA$=" "THENL=0:GOTO160
150 GOTO110
160 M=5:CLS
170 ' ┌───────────────┐
180 ' | Presentation  |
190 ' └───────────────┘
200 G=0:GB=0:IFL>49THENL=0
210 CONSOLE0,25,33,7:COLOR,,0,4:PRINT"■■ECRAN■■■■■";USING"###";L+1:PRINT"■■PIT
■■■■";MID$("     ★★★★★",M,5);"■■■0:00:00"
220 CONSOLE:FORX=0TO10:FORY=0TO7
230 Q=PEEK($BE00+L*88+X+Y*11)
240 IFQ=2THENG=G+1
250 IFQ=6THENMX=X*3:MY=Y*3
260 CURSORX*3,Y*3:ON Q+1 GOSUB 2030,2040,2050,2060,2070,2080,2090
270 NEXT:NEXT:TI$="000000"
280 CURSOR0,24:COLOR,,7,0:PRINT"████████████████████████████████████████";
290 ' ┌─────────────┐
300 ' | Mouvements  |
310 ' └─────────────┘
320 IFSY>1THENCURSORMX,MY:GOSUB2090
330 X=MX:Y=MY:X1=X:X2=X:SY=1:MUSIC"A"
340 IF(PEEK($D000+(40*(Y+3))+X+1)=0)*(GB=0)THENMY=MY+3:GOTO770
350 GETA$:IFA$=""THENCURSOR33,12:PRINT[0,4]MID$(TI$,1,1);":";MID$(TI$,3,2);":";R
IGHT$(TI$,2):GOTO350
360 ' ┌───┐
370 ' | ■ |
380 ' └───┘
390 IF(A$<>"■")+(Y<2)THEN420
400 A=PEEK($D000+(40*(Y-3))+X+1):MUSIC"A"
410 IFGB=1THENIF(A=0)+(A=120)THENMY=MY-3:GOTO770
420 IFA$<>"■"THEN480
430 ' ┌───┐
440 ' | ■ |
450 ' └───┘
460 A=PEEK($D000+(40*(Y+3))+X+1):MUSIC"A"
470 IF(A=0)+(A=$78)THENMY=MY+3:GOTO770
480 IFA$="■"THENHX=3:MUSIC"A":GOTO630
490 IFA$="■"THENHX=-3:MUSIC"A":GOTO630
500 IFA$="E"THEN1040
510 IFA$="S"THENL=L+1:GOTO580
520 IFA$="P"THENL=L-1:IFL<0THENL=49
530 IFA$="P"THEN580
540 IFA$<>"M"THEN320
550 ' ┌────────────────────┐
560 ' | affiche un tableau |
570 ' └────────────────────┘
580 M=M-1:IFM=0THENMUSIC"EEEDDCCRRRRR":GOTO100
590 CONSOLE0,24,0,33:COLOR,,,0:CLS:GOTO200
600 ' ┌─────┐
610 ' | ■ ■ |
620 ' └─────┘
630 IF(X+HX<0)+(X+HX>30)THEN320
640 B=PEEK($D000+(40*Y)+X+HX+1)
650 IF(B=$7E)+(B=$78)+(B=$D0)+(B=0)THENMX=MX+HX:GOTO770
660 IF(X+HX*2<0)+(X+HX*2>30)THEN320
670 ' ┌────────────────┐
680 ' | Pousse un pave |
690 ' └────────────────┘
700 IF(PEEK($D000+(40*Y)+X+HX)<>$4E)+(PEEK($D000+(40*Y)+X+HX*2+1)<>0)THEN320
710 CURSORX+HX,Y:GOSUB2030:CURSORX+HX*2,Y:GOSUB2070
720 IFHX=3THENX1=X+3:X2=X+6:SY=3:GOTO770
730 X1=X-6:X2=X-3:SY=2
740 ' ┌────────────────┐
750 ' | Affiche Pitman |
760 ' └────────────────┘
770 MUSIC"A":CURSORX,Y:ONGB+1GOSUB2030,2040
780 A=PEEK($D000+(40*MY)+MX+1):GB=0
790 IFA=$78THENGB=1
800 IFA=$7ETHENG=G-1
810 CURSORMX,MY
820 ONSYGOSUB2090,2100,2110
830 ' ┌────────────────┐
840 ' | Chute d'1 pave |
850 ' └────────────────┘
860 Q=0:FORX=X1TOX2STEP3:FORY=21TO3STEP-3
870 IFPEEK($D000+(40*Y)+X+1)<>0THEN920
```

```
880 IFPEEK($D000+(40*(Y-3))+X+1)=$7ETHENCURSORX,Y:GOSUB2050:GOTO910
890 IFPEEK($D000+(40*(Y-3))+X+1)=$43THENCURSORX,Y:GOSUB2070:GOTO910
900 GOTO920
910 CURSORX,Y-3:GOSUB2030:Q=1
920 NEXT:NEXT:IFQ>0THEN860
930 IFG>0THEN320
940 ' ┌──────────────┐
950 ' | Fin d'l ecran |
960 ' └──────────────┘
970 CURSOR9,12:COLOR,,7,0:PRINT"Felicitations!":CURSOR4,14
980 PRINT[7,]"Vous avez gagne en ";MID$(TI$,3,2);" m ";RIGHT$(TI$,2);" s"
990 MUSIC"CDE":FORI=1TO3000:NEXT
1000 L=L+1:M=5:GOTO200
1010 ' ┌──────────┐
1020 ' | Editeur |
1030 ' └──────────┘
1040 COLOR,,7,1:PRINT"■*** MONITEUR PITMAN ***"
1050 PRINT">";
1060 INPUT"";A$
1070 IFA$="A"THEN1180
1080 IFA$="J"THEN1310
1090 IFA$="L"THEN1390
1100 IFA$="S"THEN1530
1110 IFA$="E"THEN1670
1120 IFA$="D"THEN1760
1130 IFA$="R"THENRUN
1140 MUSIC"A":PRINTA$;" n'est pas une commande valide":GOTO1050
1150 ' ┌───────┐
1160 ' | Aide |
1170 ' └───────┘
1180 PRINT"AIDE":PRINT
1190 PRINT"----- Commandes -----"
1200 PRINT"A : Aide
1210 PRINT"J : Jouer
1220 PRINT"L : Lire
1230 PRINT"S : Sauver
1240 PRINT"E : Effacer
1250 PRINT"D : Editeur
1260 PRINT"R : Retour
1270 PRINT:GOTO1050
1280 ' ┌────────┐
1290 ' | Jouer |
1300 ' └────────┘
1310 PRINT"JOUER"
1320 PRINT"Jeu No.-";
1330 GOSUB2200
1340 M=5:CLS:GOTO200
1350 ' ┌───────┐
1360 ' | Lire |
1370 ' └───────┘
1380 REM *** ne pas taper les lignes 1360,1370.1380.1430.
     1440.1450 si vous possedez des disquettes: modifier
     egalement les commandes WOPEN. ROPEN. INPUT/T et PR
IN/T ***
1390 PRINT"LIRE":PRINT"Cassette en place ? OUI=[O]";
1400 GOSUB2150
1410 PRINT"Appuyez sur PLAY."
1420 PRINT"Je lis les ecrans...";
1430 A=$BE00:ROPEN"PIT"
1440 FORI=1TO50
1450 INPUT/T C$:FORJ=1TOLEN(C$)
1460 POKEA,VAL(MID$(C$,J,1)):A=A+1
1470 NEXT:PRINTI;",";:NEXT
1480 CLOSE:PRINT
1490 GOTO 1050
1500 ' ┌──────────────┐
1510 ' | Enregistrer |
1520 ' └──────────────┘
1530 PRINT"ENREGISTRÊR":PRINT"Cassette en place ? OUI=[O]";
1540 GOSUB2150
1550 PRINT"Appuyez sur RECORD-PLAY."
1560 PRINT"J'enregistre les ecrans...";
1570 A=$BE00:WOPEN"PIT"
1580 FORI=1TO50:C$="":FORJ=1TO88
1590 C$=C$+STR$(PEEK(A)):A=A+1
1600 NEXT
1610 PRINT/T C$:PRINTI;",";
1620 NEXT:CLOSE:PRINT
1630 GOTO 1050
1640 ' ┌──────────┐
1650 ' | Effacer |
1660 ' └──────────┘
1670 PRINT"EFFACER":PRINT"Effacer No.-";
1680 GOSUB2200
1690 PRINT"OK?=[O]";
1700 GOSUB2150
1710 FORQ=0TO87:POKE$BE00+L*88+Q,0:NEXT
1720 GOTO1050
1730 ' ┌─────────────┐
1740 ' | Visualiser |
1750 ' └─────────────┘
1760 PRINT"EDITEUR D'ECRANS":PRINT"Ecran No.-";
1770 GOSUB2200
1780 COLOR,,,1:CLS:FORX=0TO10:FORY=0TO7
1790 Q=PEEK($BE00+L*88+Y*11+X):CURSORX*3,Y*3
1800 ONQ+1GOSUB2030,2040,2050,2060,2070,2080,2090
1810 NEXT:NEXT:X=0:Y=0:HX=0:HY=0
1820 CURSOR35,1:COLOR,,7,1:PRINTL+1;
1830 HX=0:HY=0
1840 CURSORX*3,Y*3:COLOR,,7,0:PRINT" ■■■■ ";
1850 GETA$:IFA$=""THEN1850
1860 IF(A$="■")*(Y>0)THENHX=0:HY=-1:GOTO1960
1870 IF(A$="■")*(Y<7)THENHX=0:HY=1:GOTO1960
1880 IF(A$="■")*(X>0)THENHX=-1:HY=0:GOTO1960
1890 IF(A$="■")*(X<10)THENHX=1:HY=0:GOTO1960
1900 IFA$="S"THENL=L+1:IFL>49THENL=0
1910 IFA$="P"THENL=L-1:IFL<0THENL=49
1920 IF(A$="S")+(A$="P")THEN1780
1930 IFA$="M"THEN1040
1940 Q=VAL(A$):IF(Q<1)+(Q>7)THEN1960
1950 POKE$BE00+L*88+X+Y*11,Q-1
1960 Q=PEEK($BE00+L*88+X+Y*11)
1970 CURSORX*3,Y*3
1980 ONQ+1GOSUB2030,2040,2050,2060,2070,2080,2090
1990 X=X+HX:Y=Y+HY:GOTO1830
2000 ' ┌────────────────┐
2010 ' | Caractere 3*3 |
2020 ' └────────────────┘
2030 COLOR,,4,0:PRINT"    ■    ■    ";:RETURN
2040 COLOR,,7,0:PRINT" ├─┤■├─┤■├─┤";:RETURN
2050 COLOR,,6,0:PRINT" ▂ ■ ■■ ■■■";:RETURN
2060 COLOR,,2,0:PRINT"■■■■■■■■■";:RETURN
2070 COLOR,,2,0:PRINT"■■■■■■■■■";:RETURN
2080 COLOR,,7,5:PRINT"┌─ ■   ■ ─┘";:RETURN
2090 COLOR,,7,0:PRINT" ● ■(■)■ ┌┐ ";:RETURN
2100 COLOR,,7,0:PRINT"\● ■ ■)■ ┐ ";:RETURN
2110 COLOR,,7,0:PRINT" ●/■(■ ■ ┌─";:RETURN
2120 ' ┌──────────┐
2130 ' | GET"O" |
2140 ' └──────────┘
2150 INPUT"";A$:IFA$="O"THENRETURN
2160 RETURN1050
2170 ' ┌─────────────┐
2180 ' | INPUT 1-50 |
2190 ' └─────────────┘
2200 INPUT"";A$:L=VAL(A$)-1:IF(L<0)+(L>49)THENRETURN1050
2210 RETURN
2220 ' ┌────────────┐
2230 ' | S.P.TITRE |
2240 ' └────────────┘
2250 CONSOLE2,12,0,40:COLOR,,7,5:CLS
2260 FORY=3TO12
2270 CURSOR3,Y:PRINT"■   ■   ■   ■  ■ ■ ■ ■ ■";
2280 NEXT
2290 FORX=0TO4:RESTORE
2300 FORY=0TO7
2310 READMX,MY
2320 CURSOR MX+X,MY:PRINT"■■■";
2330 NEXT
2340 CURSOR6,3+X:PRINT"■";
2350 NEXT
2360 CURSOR7,3:PRINT"■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■";
2370 CONSOLE:COLOR,,7,0
2380 CURSOR8,16:PRINT"D'apres Y.ISOKAWA (OH.MZ)"
2390 FORY=18TO24
2400 READA$:CURSOR14,Y:PRINTA$;
2410 NEXT
2420 CURSOR9,20:GOSUB2050
2430 CURSOR28,20:GOSUB2050
2440 COLOR,,7,0:RETURN
2450 ' ┌────────────────┐
2460 ' | Donnees titre |
2470 ' └────────────────┘
2480 DATA3,3,3,8,12,3,13,3,19,4,26,3,26,8,32,4
2490 DATA" ┌───MENU───┐"
2500 DATA" |            |"
2510 DATA" |M:Moniteur |"
2520 DATA" |            |"
2530 DATA" | ESP:Jeu    |"
2540 DATA" |            |"
2550 DATA" └────────────┘"
2560 ' ┌───────────────────┐
2570 ' | Donnees tableaux |
2580 ' └───────────────────┘
2590 DATA"0004000000015545055551100405000011063000000155550500001000050000012000
000004155505550555004000000000040400000000512300000051322"
2600 DATA"0002005133303332051003330050510000006505555555555000060504000000415140
0000241512000023315122005211151550055555515200000001550055"
2610 DATA"5555555000000400000020003000020310125450100021352540020310343202310163
334030020555331321352333315555555555424000053354441510533"
2620 DATA"5333001053350021510533534400105335212151053351020016555500400000000064
040000001330100000010033331110333320015420000305504420005"
2630 DATA"0012450531015132011040005550140400653501402551535014014415355150233153
5510011555355500515333535021155555333050011113336520302133"
2640 DATA"3154032413441530354103415003041003151110310251555155101110000001000040
020531551401505310512015053105120550551051405500110555300"
2650 DATA"5110020055551100100600010060005330204333152334023311543130303153331300
4015133130353105101315103035133301331331230000020000000020"
2660 DATA"0020000200060002000005450000000000200000545454545450000000000050505050
5050040010040000555100400000000000411005555543410011111140410"
2670 DATA"0132303031201042050011010210506132200520233345210514031353310531013533
310501130333310141651131154233514101134333133300010004000"
2680 DATA"0000004000005001140555100014344000000140340000000130330002061004000240
5511300012000400600400111113111111004030400110323032301103"
2690 DATA"3035301103330333011000101000110001010001400055400433064003002115331333
033024013303003333100103025401505000403333300203100000105"
2700 DATA"1111100000012441420000142412440001242143210014241244100132214231005515
1241102521511111601100000000011333433343110323632401103344"
2710 DATA"4201103220333011003202400110024034001103333033010000003354515543133525
1502514354511040133545100333003330115333333441064040003350"
2720 DATA"5050512101111111111600000000015022404240150002020201504420404015020002
0201504240224015000000000100040000000054514555100044036001"
2730 DATA"0023303300100350100001020030000010505555555505050000000000000533544000
0255554400003111531100404012001114340050011633400200113033"
2740 DATA"0500111111115001340000000003605555555103240000001303240000010303240000
1303032400010303032000130303030001114640000001041400000010"
2750 DATA"4140000001054541000010040401000105505055001004040500210550505005332340
5204021334055451400031504413315410035101504331401302330013"
2760 DATA"1330554131502363331515113000004000030353111111305350000013035304000130
0003330013000032300133330333001000300600015555611111153355"
2770 DATA"0004215333500423153555042315535200231555355503115353352021553555510155
3300000000240000000001410000005503100000000051002004000010"
2780 DATA"1505500155010000001000000000610000040533333000405555550114000000201434
5555550140401111101303032301010050242016100501210110000001"
2790 DATA"0001333333304001400600055513055500000140225001201305450001010033300001
3013333333032330404000333311111113052030000130330300001300"
2800 DATA"6400001133033000010000000000155555555500000000052400000404053113331031
5411000130153114040001001130331055211030013055416015301254"
2810 DATA"3000043000033440331000313430310603103205555531004030250312040345003140
4035000311111555550000640400000000150400004015004000041500"
2820 DATA"4020130000040501300004220010010032200555100322000000041111000002461410
0002411141020241151310104115240111033251300110035514001100"
2830 DATA"3331110056665666520555555555510000400050101523000501015144005011333234
0601100033315011111111150000000444440002004222200011024222"
2840 DATA"6000052242210000022242100015222241000102222210005111111100404000000003
0111111100041400001000321002010500414050102503010001010000"
2850 DATA"0006155101511155000340040000011531111110015300001550153040010201503330
1150150323011101103330100000000061160004000211304424403110"
2860 DATA"5242425011024222420110542224501114222224111122222221111111111111001400
4000200110110015204101020461041010101200110100411010101041"
2870 DATA"0000010101020000100000103320040204033401205030344011111011330000033113
3046004011000133320110030001501303333010000000000400415555"
2880 DATA"5550021111160004204000551022320000510423320005102433320201322333305010
2200000520400000000545510004050204115550505051100200303011"
2890 DATA"0150031311101000250511000061010112500000400140004011421255540013414000
4001021233042010313003250330100012533405516125003050000000"
2900 DATA"0055155525555551535404025515554022455102021423551525555205510306002055
5555555550204100000015041000000150410000001304100040051 04"
2910 DATA"1551400500416003105422330031050231210010040051002001330410154010203101
0401033010033110006100401333331015012000000103155055505333"
2920 DATA"2421424422242213460444422111300002441500111144413451224444513313330005
5315520111111133103004333333641112241112422144210103214422"
2930 DATA"111033131113330233310133201423111331012202221210"
2940 DATA"F":REM OUF!
```

LOGICIEL « LOGITHEQUE »

La mise au point du tableur promis dans notre dernier numéro n'étant pas encore terminée, nous avons inversé l'ordre de nos parutions et vous proposons, dès maintenant, le logiciel prévu pour notre prochain bulletin.

Nous utilisons ce programme pour générer les logithèques MZ présentées dans les différents «Sharpentiers ». Il s'agit, en fait, d'une gestion de fichier spécialisée comportant Création, Annulation et Liste classée et triée des programmes. Vous pourrez ainsi constituer sur cassette ou disquette votre propre logithèque éditée sur écran ou imprimante. Dès son lancement par RUN, le programme lit automatiquement le fichier appelé « LISTE PROGS.80B » (sous-programme situé en ligne 900). Lors de sa première utilisation, il vous faudra donc supprimer provisoirement la ligne 97 qui appelle cette routine puisque votre fichier n'est pas encore créé.

LOGITHEQUE peut être aisément adapté au MZ 800 moyennant la modification de certaines commandes :
PRINT CHR$(6) devient CLS
CONSOLE C80 devient INIT « CRT :M3 ».
Il vous faudra également adapter les caractères de contrôle de l'imprimante (PRINT CHR$(X) aux caractéristiques de votre propre imprimante, le programme étant conçu pour imprimante SHARP P5.

```
 10 REM LOGITHEQUE MZ 80 B - S.B. - 11/84
 11 REM
 89 REM  Initialisations
 90 A=250:DIMTT$(A),CO$(A),CA$(A),TM(A),CT$(10),L$(4)
 95 LN$="**********":CL$="JUED"
 97 GOSUB 900
 99 REM  Menu
 100 CONSOLEC80,R,S0,24:PRINTCHR$(6):PRINTSTRING$("-",80):
 110 L$(1)="
 120 L$(2)="
 130 L$(3)="
 140 L$(4)="
 145 FORI=1TO4:PRINTL$(I):NEXT
 150 PRINTSTRING$("-",80)
 153 PRINTTAB(32);"A U  M E N U . . .":PRINT:PRINT
 160 PRINT" ** 1 ** ENTREE DE NOUVEAUX PROGRAMMES":PRINT
 170 PRINT" ** 2 ** EFFACEMENT DE PROGRAMMES":PRINT
 180 PRINT" ** 3 ** LISTE DES PROGRAMMES":PRINT
 190 PRINT" ** 4 ** ENREGISTREMENT LISTE":PRINT
 210 PRINT" ** 5 ** FIN":X=0:Y=22
 220 CURSORX,Y:PRINT"------------------------------------------------ VOTRE CHOI
X : ";
 223 GETRE$:IFRE$=""THEN223
 225 RE=VAL(RE$):IF(RE=0)+(RE>5)THENGOSUB10100:GOTO220
 230 ON RE GOTO300,500,600,800
 250 END
 300 REM  Entree nouveaux programmes
 305 NP=NP+1:TR=0:EN=1
 310 PRINTCHR$(6):CURSOR0,21:PRINTSTRING$("-",80)
 312 PRINTTAB(20);"E N T R E E  N O U V E A U  P R O G R A M M E
 313 PRINTTAB(19);"---------------------------------------------";
 315 CURSOR1,19:PRINTSIZE;" Octets libres.                                   **
 ENTREE No. ";NP;" **";CHR$(5)
 320 PRINTTAB(21);"------------------":X=1:Y=2
 330 CURSORX,Y:INPUT"TITRE DU PROGRAMME : ";TT$
 335 IF(LEN(TT$)>16)+(LEN(TT$)<2)THENGOSUB10100:GOTO330
 340 CURSOR21,4:PRINT"--------------------------------------------------------
"
 342 PRINT:PRINT:PRINT"----------------------------------------":X=7:Y=5
 350 CURSORX,Y:INPUT"COMMENTAIRES : ";CO$
 355 IFLEN(CO$)>90THENGOSUB10100:GOTO350
 357 GOSUB10900:GOSUB11100:CURSOR10,9:PRINT"CATEGORIE : ";:GOSUB10200:CA$=RE$
 405 GOSUB10900:CURSOR1,23:PRINT"En K-OCTETS, sans decimale.":X=5:Y=11
 410 CURSORX,Y:INPUT"TAILLE MEMOIRE : ";TM$
 415 TM=VAL(TM$):IF(TM=0)+(TM<>INT(TM))+(TM>64)THENGOSUB10100:GOTO410
 430 CURSOR1,22:PRINT"** POUR CONTINUER :         [ E ] E R R E U R  D A N S  L '
 E N T R E E";
 440 CURSOR1,23:PRINT"----------------------        [ N ] N O U V E A U  P R O G R
A M M E";
 450 CURSOR27,24:PRINT"[ M ]  M E N U ";:GOTO 10300
 499 REM  Annulation prog. de la liste
 500 PRINTCHR$(6):CURSOR0,21:PRINTSTRING$("-",80):TR=0:EN=1
 510 PRINTTAB(5);"A N N U L A T I O N  D ' U N  P R O G R A M M E  D E  L A  L I
 S T E"
 520 PRINTTAB(4);"------------------------------------------------------------
---------";CHR$(5)
 530 PRINT:INPUT" Titre exact du programme a annuler : ";TA$:A=0
 550 FORI=1TONP:IFTA$=TT$(I)THENA=I
 555 NEXTI
 560 IFA=0THENPRINT:PRINT"♦♦♦ PROGRAMME INCONNU DANS LA LISTE ♦♦♦":GOSUB10700:GO
TO10800
 570 GOSUB10510:PRINT
 575 PRINT"   On l'efface vraiment ? (O/N) "
 577 GETRE$:IFRE$="N"THEN100
 578 IFRE$<>"O"THEN577
 580 GOSUB10900:CURSOR20,22:PRINT" PATIENCE...: J'EFFACE ET JE REMET DE L'ORDRE"
 585 FORI=ATONP-1:TT$(I)=TT$(I+1):CO$(I)=CO$(I+1):CA$(I)=CA$(I+1):TM(I)=TM(I+1)
 590 NEXTI:NP=NP-1:GOSUB10800
 599 REM  LISTE DES PROGRAMMES
 600 PRINTCHR$(6):CONSOLES0,20:CURSOR0,21:PRINTSTRING$("-",80)
 605 GOSUB10900:CURSOR23,23:PRINT"L I S T E  D E S  P R O G R A M M E S";:CURSOR
22,24:PRINTSTRING$("-",39);CHR$(5)
 607 IFTR=0THENGOSUB1500:TR=1
 610 PRINT" ♦♦ LISTE : [E]  SUR ECRAN":PRINT:PRINTTAB(12);"[I]  SUR IMPRIMANTE"
 615 GETLP$:IFLP$=""THEN615
 617 IFLP$="I"THENGOTO700
 618 IFLP$<>"E"THENGOSUB10700:GOTO615
 619 REM  Liste sur ecran
 620 PRINT:PRINT" ♦♦ LISTE GENERALE [G] OU PAR CATEGORIES [C]
 622 GETRE$:IFRE$=""THEN622
 624 IFRE$="G"THEN630
 625 IFRE$="C"THEN640
 627 GOSUB10700:GOTO622
 629 REM  Liste generale sur ecran
 630 PRINTCHR$(6)
 632 FORI=1TOLEN(CL$):CP$=MID$(CL$,I,1)
 634 FORJ=1TONP:IFCA$(J)=CP$THENGOSUB11000
 636 NEXTJ,I:GOTO11200
 639 REM  Liste sur ecran par categories
 640 PRINT:GOSUB10900:GOSUB11100:CURSOR0,12:INPUT" ♦♦ CATEGORIE CHOISIE : ";CC$:
GOSUB10900:PRINTCHR$(6)
 642 FORJ=1TONP:IFCA$(J)=CC$THENGOSUB11000
 643 NEXTJ:GOTO11200
 700 PRINTCHR$(6):GOSUB10900:CURSOR0,21:PRINTSTRING$("-",80)
 710 CURSOR20,22:PRINT"L I S T E  D U  B U L L E T I N";:CURSOR20,23:PRINTSTRING
$("-",32);:GOSUB11500
 720 PRINT/PCHR$(21);CHR$(18);STRING$("-",40);CHR$(17):FORI=1TO4:PRINT/PL$(I):NE
XT:PRINT/PCHR$(18);STRING$("-",40):PRINT/P
 725 FORJ=1TOLEN(CL$):PRINT/P:PRINT/P:PRINT/PCHR$(21);CHR$(18);CT$(J);CHR$(18);S
TRING$("-",40):PRINT/PCHR$(19):D$=MID$(CL$,J,1)
 730 FORI=1TONP:IFD$=CA$(I)THENGOSUB11600
 735 NEXTI,J
 740 GOTO11200
 799 REM  Ecriture liste sur disquette
 800 DELETE"LISTE PROGS.80 B"
 803 WOPEN#1,"LISTE PROGS.80 B"
 805 PRINT#1,NP
 810 FORI=1TONP:PRINT#1,TT$(I),CO$(I),CA$(I),TM(I)
 820 NEXTI:CLOSE#1
 830 GOTO100
 899 REM  Lecture liste sur disquette
 900 ROPEN#1,"LISTE PROGS.80 B"
 905 INPUT#1,NP
 910 FORI=1TONP:INPUT#1,TT$(I),CO$(I),CA$(I),TM(I)
 920 NEXTI:CLOSE#1
 930 RETURN
 1500 REM  Tri avant liste
 1530 PRINT:PRINT" Patience, je trie ...":PRINT
 1540 PRINT"(";:FORI=1TONP-1:PRINT" ";:NEXT:PRINT")"
 1545 CURSOR1,4
 1550 FORI=1TONP-1:FORJ=I+1TONP
 1555 IFLEFT$(TT$(J),2)<LEFT$(TT$(I),2)GOSUB1590
 1560 NEXT:PRINT".";:NEXT:PRINT:PRINT
 1580 RETURN
 1590 TR$=TT$(I):TT$(I)=TT$(J):TT$(J)=TR$:TR$=CO$(I):CO$(I)=CO$(J):CO$(J)=TR$:TR
=TM(I):TM(I)=TM(J):TM(J)=TR:TR$=CA$(I):CA$(I)=CA$(J):CA$(J)=TR$:RETURN
 9998 REM              S O U S - P R O G R A M M E S
 10099 REM  Erreur : Ligne entree effacee
 10100 CURSORX,Y:PRINTSPACE$(150):CURSORX,Y:GOSUB10700:RETURN
 10199 REM  REPONSE TYPE DE PROG.
 10200 GETRE$:IFRE$=""THEN10200
 10210 IFRE$="J"THENPRINT"Jeu":RETURN
 10230 IFRE$="U"THENPRINT"Utilitaire":RETURN
 10250 IFRE$="E"THENPRINT"Enseignement, Scientifique":RETURN
 10260 IFRE$="D"THENPRINT"Divers":RETURN
 10270 GOSUB10700:GOTO10200
 10299 REM  Reponse choix entree progs.
 10300 GETRE$:IFRE$=""THEN10300
 10310 IFRE$="E"THENGOTO310
 10320 IFRE$="N"THENGOSUB10345:GOTO305
 10330 IFRE$="M"THENGOSUB10345:GOTO100
 10340 GOTO10300
 10345 TT$(NP)=TT$:CO$(NP)=CO$:CA$(NP)=CA$:TM(NP)=TM:RETURN
 10499 REM  Coord. prog. demande effacement
 10510 I=A:PRINT:PRINT:PRINT" ## COMMENTAIRES......";CO$(I)
 10520 PRINT:PRINT" ## CATEGORIE.........";:RE$=CA$(I):GOSUB10210
 10600 PRINT:PRINT" ## RAM NECESSAIRE....";STR$(TM(I));" K.OCTETS.":PRINT
 10630 RETURN
 10699 REM  Musique alarme
 10700 TEMPO7:MUSIC"+A3R2+AR+AR+AR"
 10710 RETURN
 10799 REM  Choix apres effacement
 10800 GOSUB10900
 10805 CURSOR0,22:PRINT"POUR CONTINUER : [N]  N O U V E L  E F F A C E M E N T"
 10810 PRINT"                    [M]  M E N U";
 10820 GETRE$:IFRE$=""THEN10820
 10830 IFRE$="N"THEN500
 10840 IFRE$="M"THEN100
 10850 GOSUB10700:GOTO10820
 10899 REM  Effacement fenetre inferieure
 10900 FORI=22TO24:CURSOR0,I:PRINTSPACE$(79);:NEXT:RETURN
 10999 REM  Impression ligne liste progs. simplifiee sur ecran
 11000 PRINT"- ";TT$(J);STRING$(".",32-LEN(TT$(J)));"cat.";CA$(J);" ||  ";TM(J);T
AB(50);"K/oct. ||"
 11050 GETRI$:IFRI$=""THENRETURN
 11060 GETRI$:IFRI$=""THEN11060
 11070 RETURN
 11099 REM  Affichage categories dans fenetre
 11100 CURSOR1,22:PRINT" [J] Jeux                 [E] Enseignement, Scientifique";
 11110 CURSOR1,23:PRINT" [U] Utilitaires          [D] Divers";
 11120 RETURN
 11199 REM  Nouvelle liste ou menu
 11200 GOSUB10900:GOSUB10700:CURSOR1,22:PRINT"**  L I S T E  T E R M I N E E  **
   [N] N O U V E L L E  L I S T E";
 11210 CURSOR1,23:PRINT"------------------------------------   [M]  M E N U";
 11220 GETRE$:IFRE$=""THEN11220
 11230 IFRE$="N"THEN600
 11240 IFRE$="M"THEN100
 11250 GOSUB10700:GOTO11220
 11499 REM  Initialisation categories / imp.
 11500 PRINTCHR$(6):PRINT"             Positionner le papier en debut de page et ta
per une touche."
 11510 GETR$:IFR$=""THEN11510
 11515 CT$(1)="***************** JEUX *****************"
 11519 CT$(2)="************** UTILITAIRES **************"
 11523 CT$(3)="**** ENSEIGNEMENT - SCIENTIFIQUES ******"
 11525 CT$(4)="***************** DIVERS *****************"
 11590 RETURN
 11599 REM  Impression liste du bulletin
 11600 PRINT/PCHR$(20);CHR$(18);TT$(I);SPACE$(17-LEN(TT$(I)));CHR$(19);CO$(I);SP
ACE$(92-LEN(CO$(I)));CHR$(18);:IFTM(I)<10THENPRINT/P" ";
 11605 PRINT/PTM(I);" K"
 11610 RETURN
```

MONITEUR LM

Voici (enfin !) sur 1800 octets, un moniteur puissant et rapide qui comblera un manque important dans le MZ 80K.

Pour entrer cet utilitaire, tapez d'abord, sous BASIC 5025, le programme de saisie ci-contre puis entrez, sous BASIC les codes HEXA du Moniteur en faisant TREEEES ! ! ! Attention de ne faire aucune erreur.

Pour exécuter V.MON après la saisie, tapez sous BASIC USR (23040) CR.

Sauvegardez immédiatement votre dur labeur en tapant :

```
5200SV.MON              -CR
LONGUEUR :      0700    -CR-
AUTOSTART :     5A00    -CR-
```

Les fonctions de V. MON sont au nombre de 9 :

AAAAL : Affiche sur 23 lignes de 8 octets le contenu HEXA de la mémoire à partir de l'adresse AAAA (Hexa)

AAAAV Idem à L, mais ajoute les caractères ASCII correspondant aux codes affichés.

AAAA:CC CC CC... : Permet d'entrer des codes Hexa en mémoire à partir de l'adresse AAAA. Les codes CC doivent impérativement comporter 2 caractères hexa et être séparés par un espace.

AAAAG : Exécute un programme LM à partir de l'adresse AAAA.

/CC CC CC... ENTRE DES CODES Hexa à partir de l'adresse 4400H et exécute automatiquement un CALL 4400H.

.CC CC CC... Idem à /CC... mais exécute un JP 4400H.

AAAASNOM : Sauvegarde un programme ayant pour titre NOM et démarrant à l'adresse AAAA. ENTRER ENSUITE LA LONGEUR Hexa du programme (Longueur) et son adresse d'exécution automatique (Autostart).

R : Lit le premier programme LM trouvé sur la cassette.

AAAAM : Permet la modification du contenu des mémoires à prtir de l'adresse AAAA. Pour sortir de ce mode, tapez « BREAK » ou « * » puis CR.

Avec ce moniteur, vous êtes maintenant paré pour vous « éclater » en LM sur votre MZ et nous envoyer vos dernières créations.

A. Morette

```
1 GOSUB 9000
10 INPUT "ADRESSE(HEXA):";AD$
20 IF AD$="*" THEN GOTO 9050
30 FOR T=1 TO 4:POKE 4095+T,ASC(MID$(AD$,T,1)):NEXT T
40 USR(18176):AD=PEEK(4176)+PEEK(4177)*256
41 PRINT "TAPEZ  *  POUR  PRENDRE  UNE  NOUVELLE  ADRESSE"
42 INPUT "CODE:";A$
43 IF A$="*" THEN 10
55 POKE 4096,48
56 POKE 4097,48
60 FOR T=1 TO LEN(A$)
80 V$=MID$(A$,T,2)
90 F$=RIGHT$(V$,1):IF (ASC(F$)<48)+(ASC(F$)>ASC("F"))
   THEN USR(62):GOTO 9040
100 POKE 4099,ASC(F$)
110 F$=LEFT$(V$,1):IF (ASC(F$)<48)+(ASC(F$)>ASC("F"))
   THEN USR(62):GOTO 9040
120 POKE 4098,ASC(F$)
130 USR(18176)
140 POKE AD,PEEK(4176)
145 AD=AD+1
150 T=T+2:NEXT T:GOTO 42
9000 T=18176
9010 READ A:IF A=256 THEN RETURN
9020 POKE T,A
9030 T=T+1:GOTO 9010
9040 PRINT "ERREUR DE SAISIE:";V$:GOTO 10
9050 PRINT "SAISIE TERMINEE"
9060 END
10000 DATA 17,00,16,205,16,4,34,80,16,201
63000 DATA 256
```

```
LISTE PROGRAMME VAMON
5900  FE 01 CC 90 59 FE 02 CC 90 59 FE 03 CC 90 59 FE
5910  04 CC 90 59 FE 05 CC 90 59 FE 06 CC 90 59 FE 07
5920  CC 90 59 FE 08 CC 90 59 FE 09 CC 90 59 FE 0A CC
5930  90 59 FE 0B CC 90 59 FE 0C CC 90 59 FE 0D CC 90
5940  59 FE 0E CC 90 59 FE 0F CC 90 59 FE 10 CC 90 59
5950  FE 17 CC 90 59 FE 18 CC 90 59 FE 19 CC 90 59 FE
5960  1A CC 90 59 FE 1B CC 90 59 FE 1C CC 90 59 FE 1D
5970  CC 90 59 FE 1E CC 90 59 FE 1F CC 90 59 FE 00 CC
5980  90 59 C3 A9 5F 00 00 00 00 00 00 00 00 00 00 00
5990  3E 20 C9 00 00 00 00 00 00 00 00 00 00 00 00 00
59A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
59B0  00 00 00 00 00 00 00 00 F5 04 87 00 D0 08 44 00
59C0  62 0E 08 44 28 44 28 00 44 CD 17 09 A1 08 5C 16
59D0  78 14 94 14 9D 16 D0 28 00 00 00 00 00 00 00 00
59E0  21 27 11 36 00 C3 12 00 9E 11 F4 CF FF 7F FF 7F
59F0  FF 7B FF 7B FF 7B FF 7F FF 7B FF 21 27 11 36 FF
5A00  CD 80 5A 11 00 10 CD 75 5E 3A 04 10 FE 4C CA A0
5A10  5D FE 47 CA 80 5B FE 4D CA D0 5B FE 3A CA 18 5C
5A20  FE 53 CA 80 5C FE 54 CA 78 5D FE 56 CA A0 5D 00
5A30  3A 00 10 FE 52 CA 00 5D FE 4C CA D0 5D FE 4D CA
5A40  D0 5B FE 2E CA 50 5F FE 2F CA 50 5F FE 56 CA C8
5A50  5D CD E0 5A C3 03 5A 00 4C 4F 4E 47 55 45 55 52
5A60  20 20 20 3A 0D 41 55 54 4F 2D 44 45 50 41 52 54
5A70  3A 0D 00 00 00 00 00 00 3E 16 CD 12 00 C9 00 00
5A80  3E 16 CD 12 00 00 11 A0 5A CD 15 00 CD 06 00 CD
5A90  06 00 CD 06 00 CD 06 00 C9 31 2E 31 20 20 20 20
5AA0  2A 2A 20 56 41 4D 4F 4E 20 31 2E 31 20 20 20 20
5AB0  20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
5AC0  20 20 20 20 20 20 20 20 2A 20 20 28 43 29 41 2E
5AD0  4D 4F 52 45 54 54 45 20 32 37 2F 34 2F 38 35 0D
5AE0  CD 3E 00 CD 09 00 C9 00 00 00 00 00 00 00 00 00
5AF0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
5B00  00 00 00 00 00 CD 40 5B DA 50 5A CD 50 5B C3 03
5B10  5A 03 5A 00 00 00 00 00 00 00 00 00 00 00 00 00
5B20  F5 C5 2A 82 10 06 08 7E CD 00 5F 00 00 CD A0 5F
5B30  05 23 3E 00 B8 C2 27 5B C3 90 5B C1 F1 C9 06 00
5B40  11 00 10 CD 10 04 22 82 10 C9 10 04 22 82 10 C9
5B50  06 16 2A 82 10 7C CD 00 5F 7D CD 00 5F 3E 3A CD
5B60  A9 5F CD 20 5B C3 E0 5D 05 3E 00 C2 52 5B C9 FF
5B70  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
5B80  CD 40 5B DA 50 5A E9 00 00 00 00 00 00 00 00 00
5B90  3A 9A 11 FE 00 28 1F 3E 20 CD A0 5F CD A0 5F 06
5BA0  08 2A 82 10 7E CD 00 59 00 00 00 00 00 00 00 00
5BB0  00 00 00 23 10 EE 22 82 10 C1 F1 C9 00 00 00 00
5BC0  18 0D 00 00 00 00 00 00 CD 40 5B DA 50 5A 18 03
5BD0  C3 C8 5B 2A 82 10 7C CD 00 5F 7D CD 00 5F 3E 3A
5BE0  CD 12 00 7E CD 00 5F 21 71 11 35 35 11 00 10 CD
5BF0  03 00 11 00 10 CD 10 04 DA 50 5A 22 82 10 3E 30
5C00  32 03 10 32 04 10 11 03 10 CD 10 04 DA 50 5A 7D
5C10  2A 82 10 77 23 C3 70 5C 3A 05 10 FE 0D CA 50 5A
5C20  CD 40 5B DA 50 5A 21 05 10 22 80 10 7E 32 86 10
5C30  23 7E 32 87 10 21 84 10 3E 30 77 23 77 11 84 10
5C40  CD 10 04 DA 50 5A 7D 2A 82 10 77 23 22 82 10 2A
5C50  80 10 23 23 23 7E FE 0D C2 29 5C 2A 82 10 7C CD
5C60  00 5F 7D CD 00 5F CD 06 00 C3 03 5A 00 00 00 00
5C70  22 82 10 C3 D3 5B 00 00 00 00 00 00 00 00 00 00
5C80  3A 05 10 FE 0D CA 50 5A CD 40 5B DA 50 5A 22 04
5C90  11 21 05 10 06 10 7E FE 0D CA A1 5C 23 10 F7 18
5CA0  58 21 05 10 11 F1 10 01 10 00 ED B0 11 58 5A CD
5CB0  15 00 00 00 00 11 00 10 CD 75 5E 11 0C 10 CD 10
5CC0  04 DA 50 5A 22 02 11 11 65 5A CD 15 00 00 00 00
5CD0  11 00 10 CD 75 5E 11 0C 10 CD 10 04 DA 50 5A 22
5CE0  06 11 21 F0 10 36 01 CD 21 00 CD 24 00 CD 09 00
5CF0  C3 03 5A 00 00 00 00 00 00 C3 50 5A 00 00 00 00
5D00  C3 50 5D 11 F1 10 CD 15 00 CD 06 00 3A 05 11 CD
5D10  00 5F 3A 04 11 CD 00 5F 3E C6 CD 12 00 2A 04 11
5D20  EB 2A 02 11 19 7C CD 00 5F 7D CD 00 5F CD 06 00
5D30  11 60 5D CD 15 00 CD 09 5E CD 1B 00 CD 12 00 CD
5D40  06 00 CD 1B 00 FE 4F C2 03 5A CD 2A 00 C3 03 5A
5D50  CD 27 00 DA 50 5A CD 09 00 C3 03 5D 00 00 00 00
5D60  56 41 4C 49 44 41 54 49 4F 4E 28 4F 2F 4E 29 3F
5D70  0D 00 00 00 00 00 00 00 CD 40 5B 21 05 10 22 80
5D80  10 2A 80 10 7E FE 0D CA 5B 5C 2A 82 10 77 23 22
5D90  82 10 2A 80 10 23 22 80 10 C3 81 5D 00 00 00 00
5DA0  3A 04 10 FE 4C C2 B0 5D 21 9A 11 36 00 C3 00 5B
5DB0  21 9A 11 36 01 C3 00 5B 3A 00 10 FE E6 00 00 00
5DC0  00 00 00 00 00 00 00 00 21 9A 11 36 01 C3 0B 5B
5DD0  21 9A 11 36 00 C3 0B 5B 00 00 C3 30 5A 00 00 00
5DE0  CD 06 00 CD 1B 00 FE 64 C8 C3 68 5B C8 C3 68 5B
5DF0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
5E00  21 75 41 22 82 12 C3 50 12 C5 D5 E5 CD B1 0F CD
5E10  A6 0D 7E 32 8E 11 22 8F 11 21 92 11 36 D0 AF 32
5E20  00 E0 32 91 11 2F 32 00 E0 3A 97 11 B7 20 2C 1E
5E30  00 16 14 CD FF 09 CD 50 0A 78 07 D2 43 5E 1D 28
5E40  15 18 F0 15 20 ED AF 32 97 11 CD FF 09 CD CA 08
5E50  FE F0 28 F6 18 0F 3E 01 32 97 11 CD CA 08 FE F0
5E60  28 CD CD FF 09 F5 CD A6 0D 3A 8E 11 2A 8F 11 77
5E70  F1 E1 D1 C1 C9 F5 C5 E5 D5 AF 32 93 11 CD 09 5E
5E80  47 E6 F0 FE C0 C2 E2 5E 78 FE CD CA 5B 08 FE C9
5E90  CA B2 5E FE CA CA B2 5E FE CB CA 7D 5E FE C8 CA
5EA0  BF 5E FE C7 CA B2 5E 3A 93 11 B7 FA C9 5E C2 B8
5EB0  5E 78 CD DC 0D C3 7D 5E 3D 32 93 11 C3 C9 5E 3A
5EC0  93 11 3C 32 93 11 C3 B1 5E 78 CD A6 0D CD B5 0D
5ED0  FE 62 C2 7D 5E 3A 93 11 E6 80 EE 80 32 93 11 C3
5EE0  7D 5E 3A 93 11 B7 FA C9 5E CA C9 5E C3 B8 5E E1
5EF0  E5 00 00 00 CD 06 00 C3 50 5A 0D 4D 41 52 43 48
5F00  C5 D5 57 CD 20 5F CD 16 5F 7A CD 29 5F CD 20 5F
5F10  CD 16 5F D1 C1 C9 01 32 5F 81 4F 0A CD 42 5F C9
5F20  CB 3F CB 3F CB 3F CB 3F C9 CB 27 CB 27 CB 27 CB
5F30  27 C9 20 21 22 23 24 25 26 27 28 29 01 02 03 04
5F40  05 06 E5 CD B1 0F 77 21 71 11 34 E1 C9 00 00 00
5F50  21 00 44 22 82 10 21 01 10 22 80 10 7E 32 86 10
5F60  23 7E 32 87 10 21 84 10 3E 30 77 23 77 11 84 10
5F70  CD 10 04 DA 50 5A 7D 2A 82 10 77 23 22 82 10 2A
5F80  80 10 23 23 23 7E FE 0D C2 59 5F 3A 00 10 FE 2E
5F90  CA 00 44 CD 00 44 C3 03 5A C4 C4 C4 C4 C4 C4 C4
5FA0  E5 21 71 11 34 E1 C9 0D 3A E5 C5 01 C6 0B 6F 26
5FB0  00 09 7E CD B1 0F 77 21 71 11 34 C1 E1 C9 00 00
5FC0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
5FD0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
5FE0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
5FF0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
6000  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```



1) Vous désirez recevoir des programmes. Faites-nous parvenir, dans ce cas, la somme de 15 F par programme demandé, ainsi qu'une cassette vierge, nous vous retournerons vos programmes sous 20 jours maximum. OFFRE LIMITEE A 3 PROGRAMMES PAR ENVOI.

2) Reportez vous à notre rubrique « INFOS » pour connaître les nouvelles modalités d'envoi et d'acquisition de logiciels.

Pour chaque programme est écrit en premier la machine sur laquelle il a été crée, puis les différentes machines sur lesquelles il est certain de fonctionner tel que. De plus il est mentionné aussi la taille mémoire en Ko pour les groupes (par ex 1260/61) comprenant des PC identiques hormis pour la Mev.

A: 1211/12 • B: 1245 • C: 1251/55 • D: 1260/61 • E: 1401/02 • F: 1350 • G: 2500

La compatibilité est normalement assurée du PC le plus ancien vers le plus récent (de A vers G) et ceci dans la limite et/ les consitions définies dans chaque manuel. A... signifie ABCDEFG par souci de clarté. Pour les compatibilités non notifiées explicitement, il reste toujours possible avec un minimum de travail d'adapter tout programme quelqu'il soit, de toute machine vers une autre. Car tous les essais faute de temps n'ont pu être efectués.

```
*********************************** JEUX *************************************

ADVENTURE         cherchez le tresor cache ds l'une des 64 sall 2.5 Ko E
APOLLO            jeu d'aventure .....                          3.5 Ko E
AWELE             version amelioree du pgm du No 11             2.5 Ko C
BANQUE            achat, revente d'actions ds un temps imparti  1.0 Ko a...
BEACH HEAD        il faut detruire la base ennemie              2.5 Ko E
BLACK JACK        jeu de cartes classique                       3.0 Ko FG
BLACK JACK        black jack en 21 pts et 4 manches             1.4 Ko E
CHALLENGER        pilotez la navette spatiale aux instruments   2.2 Ko C v.3
CHASSE S/ MARIN   recherchez en 3D un s/marin et coulez-le      2.0 Ko CDEFG
DODGE'EM          essayez de survivre sans toucher d'obstacles  3.5 Ko E
FLECHETTES        il faut toucher 3 cibles                      1.1 Ko A...
INVADERS          abattez au canon les vaisseaux ennemis        0.6 Ko A...
LABYRINTHE        essayez de sortir de ce labyrinthe            1.8 Ko F
MAX               jeu des Chiffres et des Lettres               1.4 Ko A...
MISSION SPATIAL   4 epreuves pour une mission spatiale ardue    3.0 Ko CDEFG
NAVAL             bataille navale classique contre le PC        1.4 Ko A...
OLPHOUS           jeu graphique presente dans le No 13          3.0 Ko F
PUISSANCE 4       jeu classique :5 pions, 7 colonnes            2.3 Ko CDEFG
ROULETTE          pariez sur un nbre entre 0 et 36              1.0 Ko CDFG
SCRAMBLE          jeu intergalactique,plusieurs tableaux: No 11 2.5 Ko C
SCRAMBLE III      amelioration de SCRAMBLE                      3.0 Ko C
SIMUL.ECO.        vous etes le president pour 10 ans d'un pays  3.5 Ko E
SPACE TRAINING    aux commandes du vaisseau, detruisez les cibl 2.0 Ko FG
START MOGURA      jeu lm graphique et sonore publie dans No 11  1.2 Ko E
TIR A L'ARC       5 fleches a placer ds la cible: No 13         5.0 Ko FG
WARP FACTOR       explorez l'espace a bord de votre vaisseau    2.8 Ko E

******************************** UTILITAIRES *********************************

ANA               pgm d'analyse des pgms, memoires avec conv.   3.4 Ko D
BILAN JOUR        comptabilite journaliere pour medecin         1.4 Ko A...
CHRONO            transforme votre PC en chrono: No 14          0.3 Ko E
CONV DEC HEX      conversion deci-hexa en LM                    0.3 Ko BC
DESASSEMBLEUR     adaptation du pgm paru dans le No 8           4.6 Ko D
GRAPHIC AID       permet de dessiner a l'ecran tout motif       1.3 Ko C
HEXAS             mini assembleur editeur sur k7 uniquemt       2.1 Ko C
INVIDEO           inversion video tres rapide                   ---    F
MONITEUR LM       moniteur lm publie dans le No 12              1.0 Ko C
PEINTURE          gestion de fichier                            1.4 Ko A...
RENUM             renumerote un pgm y compris les branchements  ---    BC
RENUMBER          renumerotez vos pgms entierement: No 10       0.3 Ko BC
RENUMEROTATION    renumerote les numeros de lignes              0.3 Ko F
SCROLLINGS        4 routines LM pour 4 types de scrolls         ---    F
TAROT             decompte automatique des pts au tarot         3.4 Ko C
TENNIS            arbitrage d'un match de tennis                1.0 Ko A...

******************************* ENSEIGNEMENT *********************************

BAMATH            pour les ex ou futurs bacheliers (maths stats 1.9 Ko C
C.MASSE           calcul le centre de masse d'un systeme        1.0 Ko BCDEFG
CALCUL            ensemble de pgms utiles pour les maths        3.4 Ko C
CALCULS PRECIS    permet d'effectuer des multiplications exacte 1.4 Ko A...
CHUTE LIBRE       permet l'etude de la chute libre en general   3.0 Ko FG
CINEMATIQUE       determine diverses expressions importantes    3.0 ko FG
CLASSE            divers pgms de gestion d'une classe           1.4 Ko A...
CONJ ESP          conjugaison des verbes reguliers espagnols    1.4 Ko A...
DECLIN. LATINES   apprendre et reviser les declinaisons latines 3.4 Ko CDEFGH
DERIVATEUR        calcul de la derivee d'une expression         9.0 Ko DF
DOSAGE            calcul de PH de solutions aqueuses            0.5 Ko BCDEFG
DYNAMIQUE         choix multiples de calculs de dynamique       3.0 Ko FG
ETUDE FCT         etude et trace de F(x)                        1.4 Ko A...
FACT              calcul de factorielles en multiprecision      variab BCDEFG
FONCT             recherche de fonctions a partir de pts        1.2 Ko A...
HP                simulation d'une HP                           3.5 Ko E
INTEG G           integrale definie methode de Gauss            0.7 Ko BC
INTEG S           integrale definie regle de Simpson            0.2 Ko BC
INTEGRATION NUM   integrale numerique: methode de GAUSS         3.0 Ko FG
MATHS             calculs mathematiques divers                  1.4 Ko A...
MAX FCT           calcul du maximum ou 0 d'une F(x)             0.3 Ko BC
MCQ REL           QCM sur les pronoms relatifs anglais          1.4 Ko A...
NOTEXMO           calcul notes extremes, moyennes, repartition  1.4 Ko A...
ONDE              calcul de la longueur d'onde,celerite,etc     1.0 Ko BCDEFG
PERAXO            coord cart planes d'une proj axonometrique    1.0 Ko A...
PERCEA            coord cart planes d'une proj perspective      0.8 Ko A...
POLY              calcul des valeurs numerique d'un polynome    1.4 Ko A...
POLYDIR           division de polynomes                         1.4 Ko A...
RACINE            racine carree, cubique et divers calculs      1.0 Ko A...
REGRESSIONS       4 types de regressions: lin,ln,exp,puissance  0.9 Ko A...
RELATIF           exercice sur les pronoms relatifs en anglais  1.4 Ko A...
SOLEIL            comment se reperer au Soleil                  1.0 Ko A...
STATISTIQUES      pour un nbre illimite de sections avec cumul  1.1 Ko A...
SUPERVI           pour apprendre les verbes irreguliers anglais 1.4 Ko A...
TENSE             test sur l'emploi des temps en anglais        1.4 Ko A...
TRACE DE FONCT.   avec les possibilites graphiques du 1350      3.0 Ko FG
TRAJECTOIRE       calcul de trajectoire en pesanteur uniforme   3.0 Ko FG
X MATRICE         multiplication de matrices carrees (dim 8)    1.4 Ko A...

********************************** DIVERS ************************************

ASTRO             pgm tres complet d'astronomie                 7.5 Ko D
AVINAV            pgm destine aux pilotes prives                1.4 Ko A...
BALARM            balistique des armes rayees portatives        1.3 Ko A...
BALLE             calcul des derives transv et vert d'une balle 1.4 Ko A...
BIORYTHMES 1      calcul et trace de votre biorythme            2.0 Ko FG
BIORYTHMES 2      2 pgms :calcul et graphique                   3.0 Ko FG
CALENDRIER        2 petits pgms de calculs de dates             0.7 Ko BCDEFG
CODAGE            coder ou decoder vos messages                 3.0 Ko FG
COEFBAL           coeff balistique d'une balle de petit calibre 0.6 Ko A...
EPHEMERIDES       calcul les positions des planetes             8.0 Ko D
POINT             aide pour navigateurs de plaisance,croisiere  1.4 Ko A...
SECTION           calcul les caract. geom. de sections composee 1.4 Ko A...
THERMO            evaluation thermique d'un batiment            3.2 Ko E
TIERCE            permet de creer aleatoirement des comb.       1.4 Ko A...
```

BULLETIN D'INSCRIPTION AU CLUB DES SHARPENTIERS

☐ Je m'inscris
au CLUB DES SHARPENTIERS

Je bénéficie de tous les
avantages du CLUB

Je suis abonné pour 1 AN
au BULLETIN du CLUB

Je vous joins mon règlement
☐ FRANCE : 160 F
☐ ETRANGER : 200 F

CHEQUE N° . . . . BANQUE . . . . . .

DATE SIGNATURE

n° 15

NOM . . . . PRÉNOM . . . .

ADRESSE . . . .

CODE POSTAL / VILLE . . . .

PAYS . . . .

PROFESSION . . . . ÂGE . . . .

MACHINE POSSEDÉE . . . . DEPUIS . . . .

ACHETÉE CHEZ . . . .

UTILISATION PRINCIPALE DE VOTRE MACHINE . . . .

. . . .

Club des Sharpentiers 151/153, avenue Jean-Jaurès 93307 AUBERVILLIERS CEDEX Tél : 48 34 93 44